THE COMPARISON OF AC AND DC ALTERNATIVES FOR SUB-TRANSMISSION NETWORKS

Frank Engelbrecht

Thesis presented in partial fulfilment of the requirements for the degree of Master of Science in Engineering at the University of Stellenbosch

Supervisor: Dr. H.J. Beukes

December 2001

DECLARATION

I, the undersigned, herby declare that the work contained in this thesis is my own original work, unless otherwise stated, and has not previously, in its entirety or in part, been submitted at a university for a degree.

Frank Engelbrecht

8 October 2001

SUMMARY

Recent advances in semiconductor technology extended the economic power range for DC transmission to just a few MW. Network planners need tools to compare AC and DC alternatives in order to find the best technical and economic solution for a specific network. TESAT, a software analysis tool, is developed to determine the optimum conductor and line technology for a network. Voltage regulation problems are identified and can be solved with network devices which have the potential to solve network problems more effectively and economically than ever before. PSAT, another software analysis tool developed in previous research, is used to model networks and support technologies. Hence, with the aid of TESAT and PSAT, line and support technologies are combined in an attempt to find the most effective solution in terms of cost and technical performance. This is demonstrated with the aid of a case study.

Furthermore, interfaces between PSAT and the real world are developed. This includes an extension to the input interface of PSAT that calculates the equivalent impedances of a transmission line automatically, as well as an interface to share data between ReticMaster and PSAT. A dispersed generation and support technology database is also developed as an extension to the output interface of PSAT.

OPSOMMING

Onlangse vooruitgang in halfgeleiertegnologie het tot gevolg dat GS transmissie ekonomies is vir slegs 'n paar MW. Netwerkbeplanners benodig gevolglik pakette om WS en GS alternatiewe te vergelyk vir 'n spesifieke netwerk. In hierdie tesis is 'n analitiese sagteware-pakket (TESAT) dus ontwikkel om die optimale geleier en lyntegnologie vir 'n netwerk te bepaal. Spanningsregulasie-probleme word geïdentifiseer en opgelos met netwerktoestelle wat die potensiaal het om netwerkprobleme meer doeltreffend en ekonomies as ooit tevore op te los. PSAT, 'n ander analitiese sagteware-pakket wat in vorige navorsing ontwikkel is, word dan ook gebruik om netwerke en steuningstegnologieë te modelleer. Dus word PSAT en TESAT gebruik om lyn- en steuningstegnologieë te kombineer. Die doel hiervan is om die mees doeltreffende oplossing in terme van kostes en tegniese werksverrigting te vind. Dit word met behulp van 'n gevallestudie gedemonstreer.

Verder word koppelvlakke tussen PSAT en die eksterne wêreld ontwikkel. Dit sluit in: (a) 'n uitbreiding van die intreekoppelvlak van PSAT wat die ekwivalente impedansie vir 'n transmissielyn outomaties bereken; (b) die koppelvlak om data te deel tussen PSAT en ReticMaster. 'n Verspreide generasie- en steuningstegnologie databasis is uiteindelik ook ontwikkel as 'n uitbreiding van die uittreekoppelvlak van PSAT.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AAAC	All-Aluminium-Alloy Conductor
AAC	All-Aluminium Conductor
AACSR	Aluminium-Alloy Conductor Steel-Reinforced
ABC	Aerial Bundled Conductor
AC	Alternating Current
ACAR	Aluminium Conductor Alloy-Reinforced
ACSR	Aluminium Conductor Steel-Reinforced
ALF	Annual Load Factor
ATI	Appropriate Technology Index
CIC	Capital Investment Cost
CL	Cost of Losses
DC	Direct Current
DVR	Dynamic Voltage Restorer
ELF	Energy Loss Factor
EVR	Electronic Voltage Regulator
FACTS	Flexible AC Transmission Systems
GC	Generation Cost
GMD	Geometric Mean Distance
GMR	Geometric Mean Radius
GUI	Graphical User Interface
HV	High-Voltage
IGBT	Insulated Gate Bipolar Transistor
KPI	Key Performance Indicator
LCC	Life-Cycle Cost
LV	Low-Voltage
MV	Medium-Voltage
OLP	Overhead Line Parameters
PCC	Point of Common Coupling
PCCL	Point of Common Coupling at Load
PCCR	Point of Common Coupling at Receiving end
PCCS	Point of Common Coupling at Sending end
PSAT	Power System Analysis Tool

PVM	Present Value of Money
SIL	Surge Impedance Load
SSSC	Static Synchronous Series Compensator
STATCOM	Static Synchronous Compensator
SVC	Static Var Compensator
SWER	Single Wire Earth Return
TCSC	Thyristor Controlled Series Capacitor
TESAT	Technical and Economical Software Analysis Tool
UPFC	Unified Power Flow Controller
VSC	Voltage Source Converter

ACKNOWLEDGEMENTS

I would like to convey my sincere gratitude to the following people and institutions that made this thesis possible.

Firstly, to Dr. Johan Beukes for his guidance and invaluable support.

To Nampower for their financial support and to allow me the opportunity for further studies.

To the members of the power electronics group who made it enjoyable and rewarding to be part of the group.

And of course to all my friends and family, especially my parents, for their patience, support and understanding.

Finally I would like to thank the Almighty and Heavenly Father for spiritual guidance and support.

CHAPTER 1 INTRODUCTION

A power transmission system serves a dual purpose, i.e. system interconnection and bulk energy transfer. Total power generation capacity and fuel cost can be minimised using a transmission system to pool power plants and load centres. Transmission interconnections enable one to take advantage of diversity of loads, availability of sources and fuel price in order to supply electricity to loads at minimum cost with a required reliability [B1].

There are various alternatives as far as bulk energy transfer is concerned, not all of them involving electric power transmission and an economic assessment is essential in each case. Whenever the transmission distance is sufficiently large, and restricting the choice to the electrical alternatives, the case of high voltage direct current (HVDC) transmission is well established in spite of the relatively high cost of the dual conversion required [B2]. The use of the Insulated Gate Bipolar Transistor (IGBT) in DC transmission, however, has extended the economic power range for DC transmission to just a few MW.

In less obvious decisions, such as schemes to increase the power transmission capability inside of a system or to transmit a small amount of power over a large distance, the economic comparison between alternating current (AC) and DC must include the cost of lines, terminals, any special apparatus needed for voltage support, short circuit limitation, etc. The energy losses and the plant needed to supply it must also be capitalised.

To achieve a meaningful and generally applicable comparison between AC and DC in marginal cases is indeed a very difficult task. Among the factors responsible for the complexity of a generalised theory are [B2]:

- The wide range of practical situations involving different conditions among countries, e.g. the costs of overhead lines vary from country to country by a factor as high as 2.5, while the cost of the converter terminals varies very little;
- The lack of technical comparability, given the rather different degrees of AC and AC-DC power systems;

- The need to consider the long-term effects on overall system design and cost when choosing among alternative plans for system development;
- The rapid development made in the technology of both AC and DC transmission. In this respect transmission line costs have experienced a large increase in recent years. Since the line cost is relatively lower in the case of DC transmission, this effect has affected the AC alternative. However, the appearance of Flexible AC Transmission System (FACTS) devices is now having a similar beneficial effect on AC transmission costs.

Network planners need tools to compare AC and DC alternatives in order to find the best technical and economic solution when planning new networks or upgrading existing networks. An analysis tool would provide planners with the ability to compare AC and DC transmission line technologies in a manner known to power system engineers, i.e. with loadflow analysis and financial comparisons between different line options. The two most important characteristics of this tool must be:

- Assistance in the understanding of AC and DC as a transmission technology and the problems associated with these technologies;
- Provision of an analysis tool for first assessment of the voltage regulation, line losses and life-cycle costs (LCC) of different conductors from an AC and DC perspective in order to select the optimum conductor for the task at hand.

Such an analysis tool was developed under the name TESAT (Technical and Economical Software Analysis Tool). TESAT allows network planners to evaluate different AC and DC line options on a per project basis. The optimum conductor for a particular AC or DC line technology is identified, as well as any voltage regulation problems that will require some form of line compensation. Apart from being an analysis tool, TESAT can also be used as a training tool to educate network planners in the use of AC and DC as a transmission technology.

Network technologies present an alternative in terms of network devices, which can be used to solve network problems. Recent advances in semiconductor technology pushed a large number of network devices onto the market. These devices have the potential to solve network problems, including voltage regulation, more effectively and economically than ever before [A1]. All network support devices can be classified into one of four categories according to their configuration, i.e. shunt, series, series-shunt and in-line devices. DC transmission is an in-line technology that connects two separate AC networks by a DC link. DC is, therefore, not a grid network in the way that an AC system is, nor is it expected to be. The role of DC transmission, for economic reasons, is to interconnect AC systems where a reliable AC interconnection would be too expensive [B3].

When constructing new networks, utility planners are frequently reluctant to integrate network support technologies with the planning process, mainly due to a lack of adequate planning tools. To address these needs a software analysis tool was developed to model networks and support technologies. The aim of PSAT (Power System Analysis Tool) is to assist in the understanding of network problems, network technologies and their interaction and to provide an analysis tool for first assessment in the network problem-solving process [A2].

When planning a new network or upgrading an existing network, the aim of TESAT is to assist in the selection of:

1. The line technology for the network;
2. The optimum conductor for the line technology;
3. Identification of any voltage regulation problems.

After determining the three aspects above with the aid of TESAT, PSAT can subsequently be used to analyse the complete network. Voltage regulation problems in the new networks can be identified and solved in the design phase, resulting in the construction of an already compensated network. This can result in extensive cost benefits as post-construction alternations to the network are eliminated. The economic impact of the various solutions to a network problem can be evaluated in TESAT with various conductors for a particular line technology, in order to select the most effective solution, both in technical and financial terms.

It is important to note the difference between TESAT, PSAT and other simulation packages. When designing new networks TESAT can be used to determine the optimum conductor for a particular line technology and identify any voltage regulation problems. PSAT can then be used to analyse the complete network, evaluating all possible solutions and identifying one or two optimum solutions, which can quickly be evaluated in TESAT to judge the cost implications of the various solutions to the network problem. These optimum solutions can then be studied in detail in existing packages such as PSS/U and ReticMaster.

Presently only three-phase and bipolar DC lines are considered in TESAT. Future developments in TESAT will include alternative line technologies such as SWER (single wire earth return), single-phase and phase-phase networks, as well as monopolar DC lines.

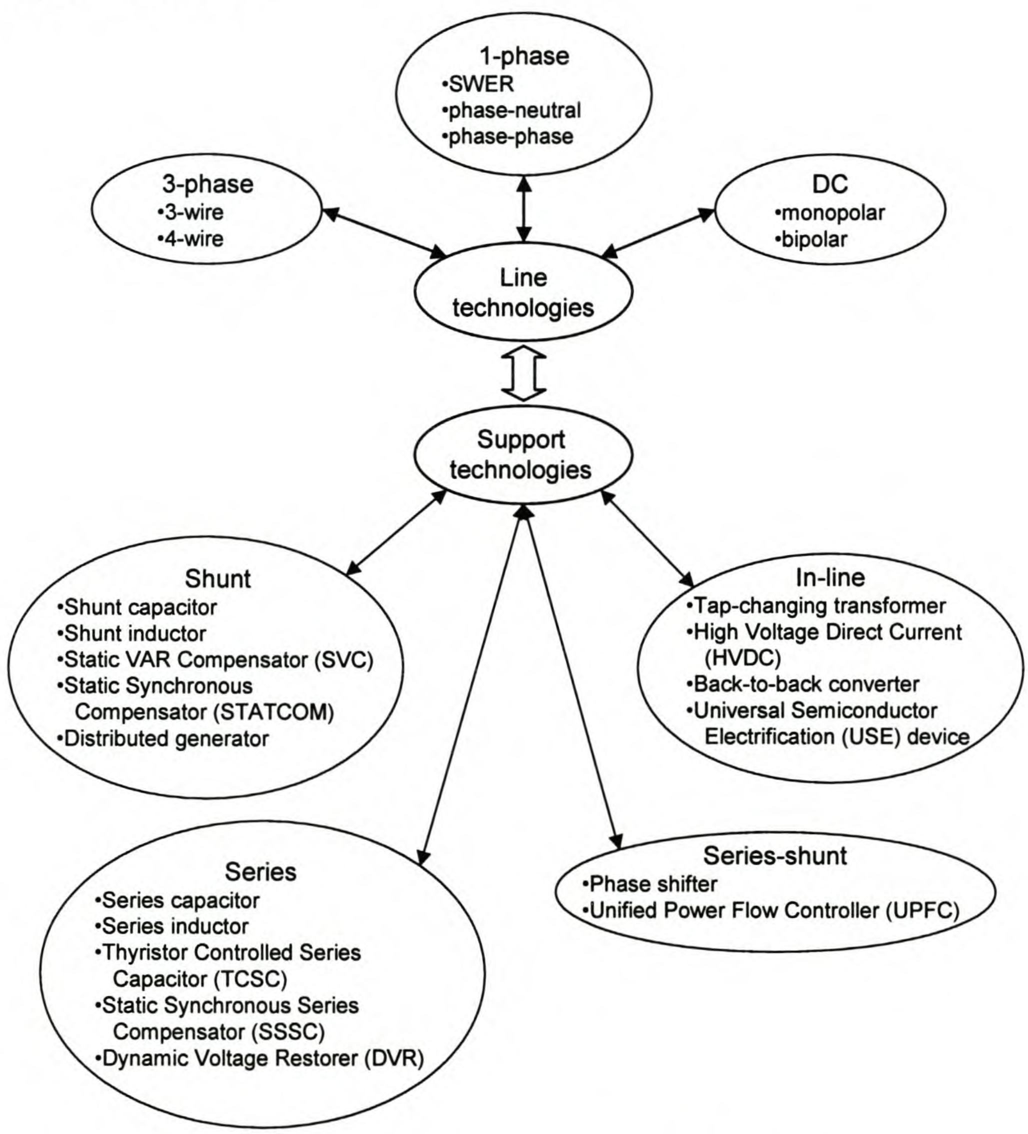


Fig. 1.1: Different line and support technologies

In today's economic environment it is of the utmost importance that optimum use is made of capital and resources when planning new networks or upgrading existing networks. The aim of this thesis is to show how the network planner can integrate line and support technologies in the planning process to determine the most effective solution in terms of cost and technical performance. This will allow the planner to select a conductor and line technology that will meet the power requirements of the network. Network support technologies can then subsequently be used to solve any voltage regulation problems in an attempt to stretch networks up to thermal capacity. The different line and support technologies are shown in Fig. 1.1. Secondly, the aim of this thesis is to document the development and practical application of the three-phase AC and bipolar DC engines of TESAT.

In the development of TESAT, the following contributions were made:

1. The process of determining the optimum conductor for different AC and DC transmission technologies was documented extensively on a high level.
2. Equations were documented to calculate the voltage regulation, line losses and transmission costs of AC transmission lines. Similar equations were developed for the voltage regulation, line losses and transmission costs of DC transmission lines.
3. A technology index was developed to compare various DC line technologies on a per project basis. The technology index for AC line technologies with and without network support devices were also documented.
4. Three-dimensional graphs were created to portray the voltage regulation, line losses, transmission costs and technology indexes of different AC and DC line technologies for all values of line position and point load.

Other contributions of this thesis include:

1. Comparisons between the load reaches and transfer capabilities of AC and DC lines at distribution voltage levels.
2. The development of interfaces between PSAT and the real world. This included the development of the line parameters interface, the ReticMaster interface and the support technology and dispersed generation database.

Chapter 2 discusses the process of determining the optimum conductor for AC and DC line technologies on a high level. The AC and DC transmission line models are discussed extensively and a software tool to calculate the circuit parameters for the

AC model is presented. Special attention is devoted to the voltage regulation, power losses and costs of AC and DC line technologies. An index is also developed to compare objectively various line technologies on a per project basis. Chapter 2 is concluded with a discussion of the TESAT interface, its features and its development.

Chapter 3 focuses on AC and DC line technologies at distribution voltage levels. The chapter starts with the deterministic and probabilistic determination of conductor ampacity ratings for both AC and DC transmission. The thermal load reach capabilities of three-phase and bipolar line technologies are calculated next at different voltage regulation margins for distribution networks. The chapter is concluded with the calculation of the transfer capabilities of these line technologies at different voltage regulation margins.

The Katima Mulilo case study is presented in Chapter 4. The study involves the construction of a new line to transmit a small amount of power over a large distance. Firstly, TESAT is used to determine the optimum conductor for a three-phase AC line. Voltage regulation of the optimised line with the transformer and load impedances is then subsequently analysed in PSAT with shunt, series and series-shunt compensators. DC transmission is implemented in PSAT as an in-line technology where TESAT determines the optimum conductor to be used for the DC link. Critical evaluation of the voltage regulation, power losses and transmission costs graphs also demonstrates the tutoring ability of TESAT. The chapter is concluded with a discussion of the most effective solution in terms of cost and technical performance.

Chapter 5 discusses the development of a PSAT interface to extract the electrical characteristics of a reticulation network from ReticMaster. The development of a support technology and dispersed generation database is also presented. Future developments for the overhead line parameters interface of PSAT and TESAT is proposed, as well as possibilities for future studies.

The thesis is concluded in Chapter 6 with a summary of the theoretical and practical results obtained with PSAT and TESAT.

CHAPTER 2 METHOD OF ANALYSIS

2.1 INTRODUCTION

This chapter discusses the development of methods for technical and economic comparisons between AC and DC transmission. The AC transmission line model is discussed first, after which section 2.3 deals with the DC transmission line model. In sections 2.4 and 2.5 the voltage regulation and line losses for AC and DC lines are discussed, followed by a discussion on transmission costs in section 2.6. The ATI models for AC and DC line technologies are presented in section 2.7. A discussion of TESAT concludes the chapter.

Transmission lines transmit electrical power from a sending end to a receiving end without supplying any consumers. In contrast, a distribution line supplies consumers directly at short intervals along the line. In practice the distinction between transmission and distribution is not so clear-cut as several large consumers are now being supplied at 132 kV and above [B4]. For the purposes of this discussion transmission voltages are assumed to be above 132 kV, sub-transmission voltages between 132 kV and 33 kV and distribution voltages 33 kV and less.

Low-voltage (LV) distribution networks operate up to and including voltages of a 1 000 V r.m.s., while medium-voltage (MV) distribution networks operate at voltages exceeding 1 000 V r.m.s. up to and including 33 kV. High-voltage (HV) networks operate at voltages above 33 kV.

2.2 THE AC TRANSMISSION LINE MODEL

Fig. 2.1 represents a line section of length Δx. Conductor spacing, types, and sizes determine the series impedance and shunt admittance. Series impedance affects line-voltage drops, I^2R losses and stability limits. Shunt admittance, primarily capacitive, affects line-charging currents, which inject reactive power into the power system [B4].

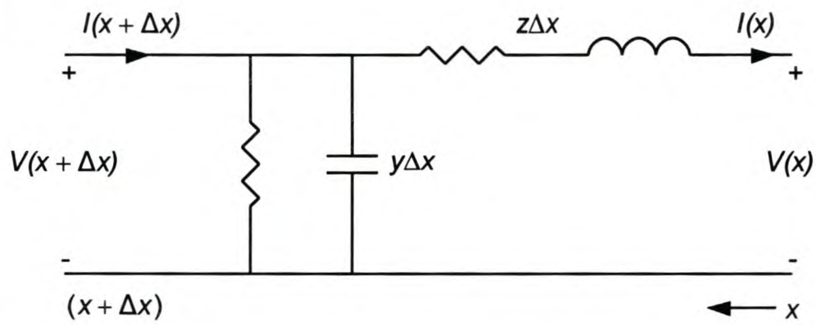


Fig. 2.1: Line section of length Δx [B5]

The matrices to calculate the series impedance and shunt admittance for untransposed as well as transposed three-phase lines with neutral conductors are developed in sections 2.2.1 and 2.2.2, followed by a discussion on the transmission line models for short, medium and long lines in section 2.2.3. Section 2.2 is concluded with a discussion on the line parameter interface of PSAT.

2.2.1 Series impedance

Series resistance

The content of this section is based on work done by Smit [C3].

The total AC resistance (R_{ac}) of a conductor can be calculated as [B5]:

$$R_{ac} = \frac{P_{loss}}{|I|^2} \quad [\Omega] \tag{2-1}$$

where P_{loss} is the conductor real power loss and I is the rms conductor current. However, many times P_{loss} and I are unknown and must be calculated. The manufacturers of overhead conductors supply only the DC resistance per kilometre (r_{dc}) at 20°C, which includes the spiralling effect for stranded conductors. The AC resistance per kilometre (r_{ac}) can be calculated in one of two ways and will be discussed next.

Method 1: Various effects

r_{ac} can be calculated as follows:

a) Obtain r_{dc} at 20°C from the data sheets for overhead aluminium conductors.

b) r_{dc} is then corrected to the required operating temperature, e.g. at 40°C.

c) The new r_{dc} is corrected for the skin effect.

d) The resistance of aluminum conductor, steel-reinforced (ACSR) conductors should be further corrected for hysteresis and eddy current losses.

e) r_{ac} is also dependent on the proximity effect of conductors.

Method 2: Empirical formula

It is difficult to calculate all the effects (c) – (e) in method 1. An alternative method can be used:

f) Empirical formulae to calculate r_{ac}.

Steps (a) – (f) will now be discussed in detail in the following sections. All-aluminium-alloy conductor (AAAC) and all-aluminium conductor (AAC) resistances should be calculated according to Method 1, with only the skin effect taken into account. ACSR conductor resistances should be calculated according to Method 2, with hysteresis and eddy current losses taken into account.

a) r_{dc} at 20°C can be obtained from the Aberdare tables [C4].

b) The variation of temperature is considered as linear over the operating range, although it is not linear over a wide range of temperatures. r_{ac} can only be calculated if r_{dc} is known at the operating temperature. The following method is used:

$$R_2 = R_1 + \alpha R_1 (T_{av} - 20) \qquad (2\text{-}2)$$

where R_1 = DC resistance at temperature T_1 [°C]

R_2 = DC resistance at temperature T_2 [°C]

α	=	temperature coefficient of resistance at 20°C
T_{av}	=	average temperature [°C]

The manufacturers of overhead conductors supply the α values for different conductors, given in Table 2-1 [C4]. There is a difference between the core and surface temperatures of a conductor, but this is only between 0.2°C and ±4°C. The average temperature, T_{av}, is therefore assumed the same as the surface temperature.

Table 2-1: Temperature coefficient of resistance at 20°C [C4]

Conductor type	α
Copper	0.00393
Cadmium copper	0.00300
Hard-drawn aluminium	0.00403
Aluminium alloy	0.00360

c) The resistance of non-magnetic conductors varies with frequency. This is caused by current flowing nearer to the outer surface of the conductor as a result of non-uniform flux distribution in the conductor. This phenomenon is commonly known as the skin effect and must be taken into account. r_{ac} at the desired operating temperature is calculated to include the skin effect as follows:

$$r_{ac} = Kr_{dc} \quad [\Omega/\text{km}] \tag{2-3}$$

where	r_{ac}	=	AC resistance at the desired frequency
	r_{dc}	=	DC resistance at the desired temperature
	K	=	value in Table 2-2

In Table 2-2, K is a function of X, where K is interpolated for a specific X value:

$$X = 0.050138\sqrt{\frac{\mu f}{r_{dc}}} \tag{2-4}$$

where	f	=	frequency [Hz]
	μ	=	relative permeability (1 for non-magnetic materials)

Table 2-2: Skin effect table [B6]

X	K	X	K	X	K	X	K
0.0	1.00000	1.0	1.00519	2.0	1.07816	3.0	1.31809
0.1	1.00000	1.1	1.00758	2.1	1.09375	3.1	1.35102
0.2	1.00001	1.2	1.01071	2.2	1.11126	3.2	1.38504
0.3	1.00004	1.3	1.01470	2.3	1.13069	3.3	1.41999
0.4	1.00013	1.4	1.01969	2.4	1.15207	3.4	1.45570
0.5	1.00032	1.5	1.02582	2.5	1.17538	3.5	1.49202
0.6	1.00067	1.6	1.03323	2.6	1.20056	3.6	1.52879
0.7	1.00124	1.7	1.04205	2.7	1.22753	3.7	1.56587
0.8	1.00212	1.8	1.05240	2.8	1.25620	3.8	1.60314
0.9	1.00340	1.9	1.06440	2.9	1.28644	3.9	1.64051

d) The resistance of ACSR conductors is affected by the hysteresis in the steel core, where the number of layers plays a role. The current spirals around the core, following the strands. The magnetising effect is opposite for the layers in a two-layer conductor and results in a small increase of the effective resistance. The biggest increase is for a single layer, up to 15%. The resistance for a three-layer conductor increases by 5% to 10% at the rated full load of the conductor.

Eddy currents are produced due to cyclic magnetisation and cause heating. The heating effect is usually negligible for non-ferrous conductors. The total heat gain of the conductor consists of an eddy current and a hysteresis component, where the hysteresis component is about 2 – 5 times the eddy current component. These values are difficult to determine and the empirical formula of (f) offers an easier solution.

e) The proximity effect is negligible for:

$$\frac{s}{d} > 5 \tag{2-5}$$

where s is the spacing between the conductors and d the overall diameter of the conductor [A3]. For a ratio s/d = 3, the proximity effect causes an increase in resistance of less than 1% [C5]. Also, the spacing between phases is usually more than 5 times the conductor diameter, which means that the proximity effect may be ignored.

f) An empirical formula is presented as an alternative method to calculate the skin, hysteresis and proximity effects [A4]. The theory works on the basis that the power input must be the same for both AC and DC circuits for the same average temperature of the conductor:

$$I_{ac}^2 r_{ac} = I_{dc}^2 r_{dc}$$
$$\Rightarrow r_{ac} = \left(\frac{I_{dc}}{I_{ac}}\right)^2 r_{dc} \qquad (2\text{-}6)$$

The DC current that will result in a certain temperature being reached is calculated and the empirical formulae are then used to convert the DC to the AC current:

i) For ACSR conductors with three layers of aluminium strands:

$$I_{ac} = I_{dc} \Big/ \sqrt{\left(1.0123 + 2.319 \cdot 10^{-5} I_{dc}\right)} \qquad (2\text{-}7)$$

ii) For ACSR conductors with single- and double-layer aluminium strands and a nominal aluminium area of ≥ 175 mm^2:

$$I_{ac} = I_{dc} \Big/ \sqrt{\left(1.0045 + 0.09 \cdot 10^{-6} I_{dc}\right)} \qquad (2\text{-}8)$$

iii) For other ACSR conductors with single- and double-layer aluminium strands and a nominal aluminium area of <175 mm^2:

$$I_k = \frac{I_{dc}}{a} \qquad (2\text{-}9)$$

where a is the nominal aluminium area and I_k the DC current per mm^2.

If $I_k \leq 0.742$ then

$$I_{ac} = I_{dc} \qquad (2\text{-}10)$$

If $0.742 < I_k \leq 2.486$ then

$$I_{ac} = I_{dc} \Big/ \sqrt{\begin{pmatrix} 1 + 0.5142 - 2.678 I_k + 5.772 I_k^2 - 6.69 I_k^3 + \\ 4.53 I_k^4 - 1.7946 I_k^5 + 0.3862 I_k^6 - 0.03488 I_k^7 \end{pmatrix}} \qquad (2\text{-}11)$$

If $2.486 < I_k \leq 3.908$ then

$$I_{ac} = I_{dc} \Big/ \sqrt{\begin{pmatrix} 1 + 0.0596 - 0.4404 I_k + 0.4974 I_k^2 - \\ 0.2328 I_k^3 + 0.0595 I_k^4 - 0.0083 I_k^5 + 0.0005 I_k^6 \end{pmatrix}} \qquad (2\text{-}12)$$

If $I_k > 3.908$ then

$$I_{ac} = \frac{I_{dc}}{\sqrt{1.1}} \qquad (2\text{-}13)$$

For bundled conductors the effective resistance per unit length of the bundled conductor is equal to r_{ac}/n, where n is the amount of conductors per bundle (n = 2, 3, or 4).

Line resistance is dependent on the chosen operating temperature. The system planner uses a single value of line resistance in all power network simulations. For clearance purposes, most of the lines are designed for an operating temperature of only 50°C. Probability studies will show that the conductor is operated at temperatures below 50°C for most of the time. Line resistances should, therefore, be calculated at an operating temperature ≤ 50°C. It is recommended that all resistances be calculated at the temperature of 40°C.

Series inductance

The line inductance depends on the partial flux linkages within the conductor cross-section and external flux linkages [B7]. For a completely transposed three-phase, three-wire line with equal phase spacing, the inductance per phase is:

$$L_1 = L_2 = 2 \times 10^{-4} \ln \frac{D_{eq}}{D_S} \quad [\text{H/km}] \qquad (2\text{-}14)$$

where

$$GMD = D_{eq} = \sqrt[3]{d_{12} d_{23} d_{31}} \qquad (2\text{-}15)$$

is the geometric mean distance (*GMD*) between the conductors of phases *a*, *b*, and *c*. In the above equation D_S is the geometric mean radius (*GMR*) of the phase conductors.

To calculate $L_1 = L_2$ for bundled conductors with bundle spacing *d*, an equivalent conductor with $GMR = D_{SL}$ replaces the bundle. D_{SL} replaces D_S in equation (2-14) and is calculated as follows:

Two-conductor bundle (twin):

$$D_{SL} = \sqrt{D_S d} \tag{2-16}$$

Three-conductor bundle (triple):

$$D_{SL} = \sqrt[3]{D_S d^2} \tag{2-17}$$

Four-conductor bundle (quad):

$$D_{SL} = 1.091\sqrt[4]{D_S d^3} \tag{2-18}$$

Series impedance

In this section equations are developed for calculation of the series impedance of the three-phase line shown in Fig. 2.2.

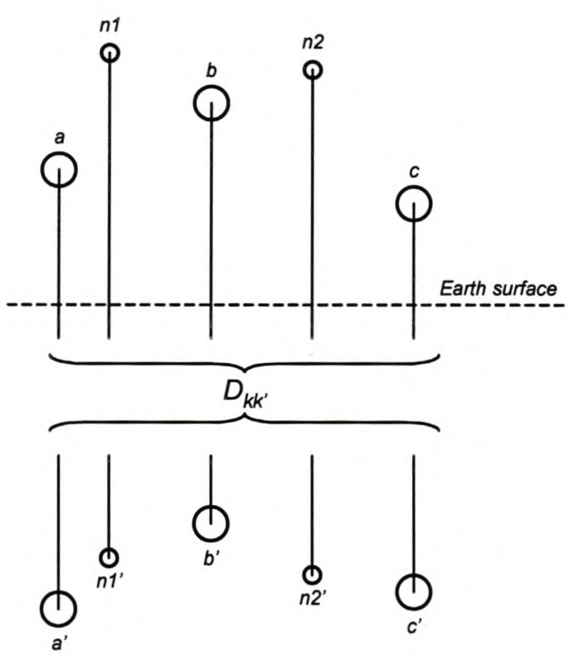


Fig. 2.2: Three-phase transmission line with earth replaced by earth return conductors

The line has three phase conductors *a*, *b*, and *c*, where bundled conductors, if any, have already been replaced by the equivalent conductors, described in equations

(2-16) - (2-18). The line also has two neutral conductors. All neutral conductors are connected in parallel and are grounded to the earth at regular intervals along the line. Any isolated neutral conductors that carry no current are omitted. The phase conductors are isolated from each other and from earth [B5].

If the phase currents are not balanced, there may be a return current in the grounded neutral wires and in the earth. The earth return current will spread out under the line, seeking the lowest impedance return path. A classic paper by Carson [A5], later modified by others [B8], [B9], shows that the earth can be replaced by a set of equivalent "earth return" conductors located directly under the overhead conductors, as shown in Fig. 2.2. Each earth return conductor carries the negative of its conductor current, has a *GMR* denoted $D_{k'k'}$, distance $D_{kk'}$ from its overhead conductor, and resistance $r_{k'}$ given by:

$$D_{k'k'} = D_{kk} \quad [\mathrm{m}] \tag{2-19}$$

$$D_{kk'} = 658.5\sqrt{\frac{\rho}{f}} \quad [\mathrm{m}] \tag{2-20}$$

$$r_{k'} = 9.869 \times 10^{-4} \times f \quad [\Omega/\mathrm{km}] \tag{2-21}$$

where ρ is the earth resistivity and f the frequency. The value of ρ for various earth types is given in Table 2-3. It is customary to use an average value of 100 Ωm if the actual earth resistivity is unknown.

Table 2-3: Resistivity of different soils [B10]

Ground Type	ρ [Ωm]
Seawater	0.01-1.0
Wet Organic Soil	10
Moist Soil	100
Dry Soil	1000
Bedrock	10^4
Pure Slate	10^7
Sandstone	10^9
Crushed Rock	1.5×10^8

All the earth return conductors are the same distance $D_{kk'}$ from their overhead conductors and have the same resistance $r_{k'}$. A circuit representation of the line is shown in Fig. 2.3.

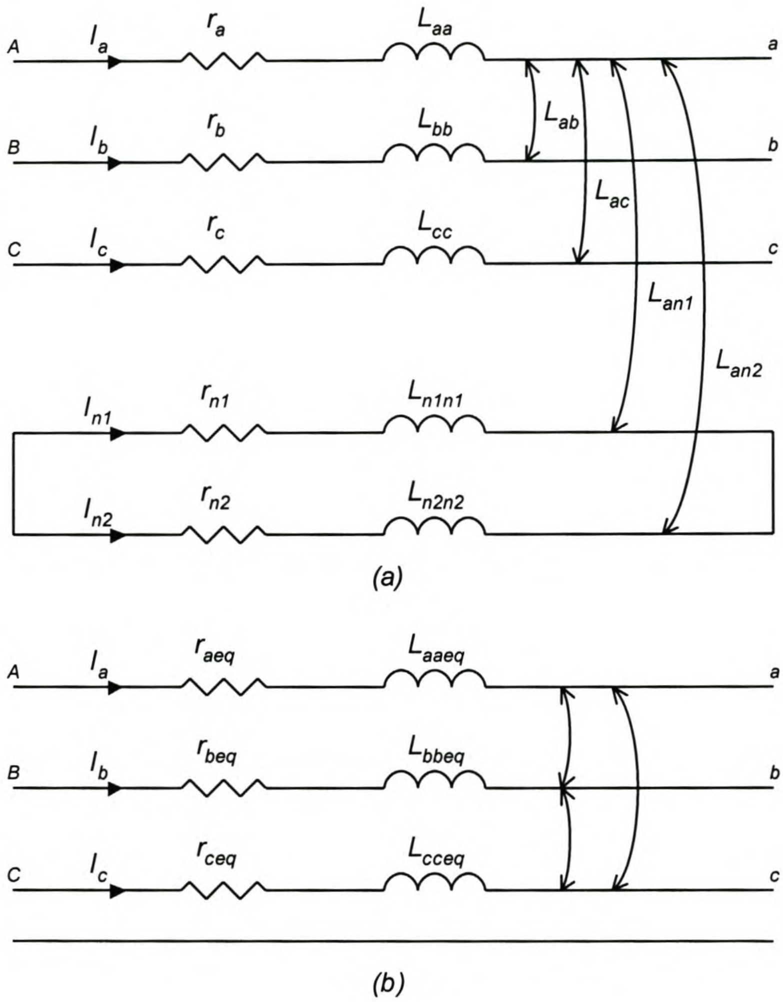


Fig. 2.3: (a) Complete circuit and (b) simplified circuit representation of series-phase impedances

Using the circuit in Fig. 2.3 (b) the vector of voltage drops across the conductors is:

$$\begin{bmatrix} V_{Aa} \\ V_{Bb} \\ V_{Cc} \\ \hline 0 \\ 0 \end{bmatrix} = \left[\begin{array}{ccc|cc} z_{aa} & z_{ab} & z_{ac} & z_{an1} & z_{an2} \\ z_{ba} & z_{bb} & z_{bc} & z_{bn1} & z_{bn2} \\ z_{ca} & z_{cb} & z_{cc} & z_{cn1} & z_{cn2} \\ \hline z_{n1a} & z_{n1b} & z_{n1c} & z_{n1n1} & z_{n1n2} \\ z_{n2a} & z_{n2b} & z_{n2c} & z_{n2n1} & z_{n2n2} \end{array}\right] \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ \hline I_{n1} \\ I_{n2} \end{bmatrix} \qquad (2\text{-}22)$$

The diagonal elements of this matrix are:

$$z_{kk} = r_k + r_{k'} + j\omega\, 2\times10^{-4} \ln\frac{D_{kk'}}{D_{kk}} \quad [\Omega/\mathrm{km}] \qquad (2\text{-}23)$$

and the off-diagonal elements, for $k \neq m$, are:

$$z_{km} = r_{k'} + j\omega 2\times10^{-4} \ln\frac{D_{km'}}{D_{km}} \quad [\Omega/\text{km}] \tag{2-24}$$

where

$$k = a, b, c, n1, n2$$
$$m = a, b, c, n1, n2$$

For equations (2-23) and (2-24) r_k is the AC resistance at 40°C and D_{kk} the *GMR* of conductor *k*. $D_{km'}$ is the distance between overhead conductor *k* and earth return conductor *m*, and D_{km} the distance between overhead conductor *k* and *m*.

Next, (2-22) is partitioned as shown above to obtain:

$$\left[\begin{array}{c}\mathbf{E_P}\\ \hline \mathbf{0}\end{array}\right] = \left[\begin{array}{c|c}\mathbf{Z_A} & \mathbf{Z_B}\\ \hline \mathbf{Z_C} & \mathbf{Z_D}\end{array}\right]\left[\begin{array}{c}\mathbf{I_P}\\ \hline \mathbf{I_n}\end{array}\right] \tag{2-25}$$

$\boldsymbol{E_P}$ is the three-dimensional vector of voltage drops across the phase conductors; $\boldsymbol{I_P}$ is the three-dimensional vector of phase currents and $\boldsymbol{I_n}$ is the two-dimensional vector of neutral currents.

Equation (2-25) is rewritten as two separate matrix equations:

$$\mathbf{E_P} = \mathbf{Z_A I_P} + \mathbf{Z_B I_n} \tag{2-26}$$

$$\mathbf{0} = \mathbf{Z_C I_P} + \mathbf{Z_D I_n} \tag{2-27}$$

Solving (2-27) for $\boldsymbol{I_n}$ and substituting back into (2-26):

$$\mathbf{E_P} = \left(\mathbf{Z_A} - \mathbf{Z_B Z_D^{-1} Z_C}\right)\mathbf{I_P} \tag{2-28}$$

or

$$\mathbf{E_P} = \mathbf{Z_P I_P} \tag{2-29}$$

where

$$\mathbf{Z_P} = \mathbf{Z_A} - \mathbf{Z_B Z_D^{-1} Z_C} \tag{2-30}$$

$\boldsymbol{Z_P}$ given by (2-30) is a 3 x 3 series-phase impedance matrix, whose elements are denoted:

$$\mathbf{Z_P} = \begin{bmatrix} Z_{aaeq} & Z_{abeq} & Z_{aceq} \\ Z_{baeq} & Z_{bbeq} & Z_{bceq} \\ Z_{caeq} & Z_{cbeq} & Z_{cceq} \end{bmatrix} \tag{2-31}$$

These series impedances are for an untransposed line; if symmetrical component transformations are applied, the results will show mutual coupling between the sequence components [B5], [B8]. Since this is not desirable, the use of symmetrical components is not advised, unless it is assumed that the line is perfectly transposed [A6]. The equations for the transposed line are simply obtained by averaging the diagonal impedances and the off-diagonal impedances:

$$\hat{\mathbf{Z}}_{\mathbf{P}} = \begin{bmatrix} \hat{Z}_S & \hat{Z}_M & \hat{Z}_M \\ \hat{Z}_M & \hat{Z}_S & \hat{Z}_M \\ \hat{Z}_M & \hat{Z}_M & \hat{Z}_S \end{bmatrix} \tag{2-32}$$

where

$$\hat{Z}_S = \frac{1}{3}\left(Z_{aaeq} + Z_{bbeq} + Z_{cceq}\right) \tag{2-33}$$

$$\hat{Z}_M = \frac{1}{3}\left(Z_{abeq} + Z_{bceq} + Z_{caeq}\right) \tag{2-34}$$

Symmetrical component transformation will give:

$$\hat{\mathbf{Z}}_{\mathbf{S}} = \begin{bmatrix} \hat{Z}_0 & 0 & 0 \\ 0 & \hat{Z}_1 & 0 \\ 0 & 0 & \hat{Z}_2 \end{bmatrix} \tag{2-35}$$

where

$$\hat{Z}_0 = \hat{Z}_S + 2\hat{Z}_M \tag{2-36}$$

$$\hat{Z}_1 = \hat{Z}_2 = \hat{Z}_S - \hat{Z}_M \tag{2-37}$$

These equations give the results that are generally desired for most purposes, but on occasion it is preferable to use the untransposed impedances in terms of the *abc* variables instead of the symmetrical component variables. Of course, in the balanced three-phase case, the positive-sequence resistance is just the resistance of the phase conductor, and the positive sequence reactance is readily calculated from (2-14).

2.2.2 *Shunt admittance*

Shunt conductance

Conductance accounts for power loss between conductors or between conductors and ground. This power loss is due to leakage currents along insulator strings and corona. In overhead lines its effect is small and usually neglected [B11].

Shunt capacitance

Firstly, a three-phase, three-wire line with equal phase spacing is considered. The effect of the earth and the neutral conductors is neglected and due to the symmetry only one phase is considered. The positive sequence capacitance, which also equals the negative sequence capacitance, can be calculated as follows:

$$C_1 = C_2 = \frac{2\pi\varepsilon}{\ln\left(D_{eq}/r\right)} \times 1000 \quad [\mathrm{F/km}] \tag{2-38}$$

where r is the outside radius of the conductor, D_{eq} is the *GMD* and ε is the permittivity of the dielectric medium (for air: 8.854×10^{-12}). In the equation above r is assumed to be the outside radius of the conductor, thus treating the stranded conductor as a solid cylindrical conductor. This assumption is valid since most of the charge on a conductor resides on its surface, and thus most of the current will only flow on the surface (skin effect).

To calculate $C_1 = C_2$ for bundled conductors with bundle spacing d, r is replaced by the modified *GMR* (D_{SC}) of the bundle, calculated as follows:

Two-conductor bundle (twin):

$$D_{SC} = \sqrt{rd} \tag{2-39}$$

Three-conductor bundle (triple):

$$D_{SC} = \sqrt[3]{rd^2} \tag{2-40}$$

Four-conductor bundle (quad):

$$D_{SC} = 1.091\sqrt[4]{rd^3} \tag{2-41}$$

To determine the zero sequence capacitance, C_0, the effect of neutral conductors and the earth must be considered [B5]. The neutral conductors and the earth also affect the positive- and negative-sequence capacitance, since they alter the electric field. Equations are subsequently developed for the phase capacitance and sequence capacitance matrices of three-phase lines with neutral conductors.

The earth surface is approximated as a perfectly conducting horizontal plane, even though the earth under the line may have irregular terrain and resistance. The effect of the earth plane is accounted for by the *method of images*, where an image conductor, which has the same radius as the original conductor, replaces the earth. This conductor lies directly below the original conductor and has an equal quantity of negative charge.

Fig. 2.4 shows a three-phase line with neutral conductors and with the earth plane replaced by image conductors. D_{km} is the distance between overhead conductor k and m and D_{kk} is the outside radius of overhead conductor k (r_k). H_{km} is the distance between overhead conductor k and image conductor m. Since all neutral conductors are grounded to the earth, the voltage between these conductors and the earth will be zero.

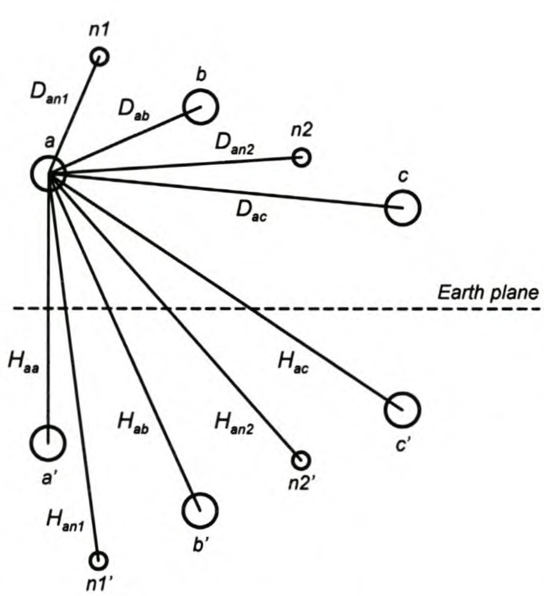


Fig. 2.4: Three-phase line with neutral conductors and with earth plane replaced by image conductors

In matrix format the voltage between the phase conductors and the earth, and the voltage between the neutral conductors and the earth, are:

$$\begin{bmatrix} V_{an} \\ V_{bn} \\ V_{cn} \\ \hdashline 0 \\ 0 \end{bmatrix} = \left[\begin{array}{ccc:cc} P_{aa} & P_{ab} & P_{ac} & P_{an1} & P_{an2} \\ P_{ba} & P_{bb} & P_{bc} & P_{bn1} & P_{bn2} \\ P_{ca} & P_{cb} & P_{cc} & P_{cn1} & P_{cn2} \\ \hdashline P_{n1a} & P_{n1b} & P_{n1c} & P_{n1n1} & P_{n1n2} \\ P_{n2a} & P_{n2b} & P_{n2c} & P_{n2n1} & P_{n2n2} \end{array}\right] \begin{bmatrix} q_a \\ q_b \\ q_c \\ \hdashline q_{n1} \\ q_{n2} \end{bmatrix} \tag{2-42}$$

The elements of the 5 x 5 ***P*** matrix are:

$$P_{km} = \frac{1}{2\pi\varepsilon} \ln \frac{H_{km}}{D_{km}} \quad [\text{F/m}] \tag{2-43}$$

where

$$k = a, b, c, n1, n2$$
$$m = a, b, c, n1, n2$$

Equation (2-42) is partitioned as shown above to obtain:

$$\left[\begin{array}{c} \mathbf{V_P} \\ \hline \mathbf{0} \end{array}\right] = \left[\begin{array}{c|c} \mathbf{P_A} & \mathbf{P_B} \\ \hline \mathbf{P_C} & \mathbf{P_D} \end{array}\right] \left[\begin{array}{c} \mathbf{q_P} \\ \hline \mathbf{q_n} \end{array}\right] \tag{2-44}$$

$\boldsymbol{V_P}$ is the three-dimensional vector of phase-to-neutral voltages; $\boldsymbol{q_P}$ is the three-dimensional vector of phase-conductor charges and $\boldsymbol{q_n}$ is the two-dimensional vector of neutral-conductor charges.

Equation (2-44) is rewritten as two separate equations:

$$\mathbf{V_P} = \mathbf{P_A q_P} + \mathbf{P_B q_n} \tag{2-45}$$

$$\mathbf{0} = \mathbf{P_C q_P} + \mathbf{P_D q_n} \tag{2-46}$$

Equation (2-46) is solved for $\boldsymbol{q_n}$, which is used in (2-45) to obtain:

$$\mathbf{V_P} = \left(\mathbf{P_A} - \mathbf{P_B P_D^{-1} P_C}\right)\mathbf{q_P} \tag{2-47}$$

or

$$\mathbf{q_P} = \mathbf{C_P V_P} \tag{2-48}$$

where

$$\mathbf{C_P} = \left(\mathbf{P_A} - \mathbf{P_B P_D^{-1} P_C}\right)^{-1} \quad [\text{F/m}] \tag{2-49}$$

$\boldsymbol{C_P}$ given by (2-49) is the 3 x 3 matrix of phase capacitances whose elements are denoted:

$$\mathbf{C_P} = \begin{bmatrix} C_{aa} & C_{ab} & C_{ac} \\ C_{ba} & C_{bb} & C_{bc} \\ C_{ca} & C_{cb} & C_{cc} \end{bmatrix} \quad [\mathrm{F/m}] \qquad (2\text{-}50)$$

In general, $\boldsymbol{C_P}$ does not satisfy the conditions for a symmetrical capacitance matrix. However, if the line is completely transposed, the diagonal and off-diagonal elements of $\boldsymbol{C_P}$ are averaged to obtain:

$$\hat{\mathbf{C}}_\mathbf{P} = \begin{bmatrix} \hat{C}_S & \hat{C}_M & \hat{C}_M \\ \hat{C}_M & \hat{C}_S & \hat{C}_M \\ \hat{C}_M & \hat{C}_M & \hat{C}_S \end{bmatrix} \quad [\mathrm{F/m}] \qquad (2\text{-}51)$$

where

$$\hat{C}_S = \frac{1}{3}\left(C_{aa} + C_{bb} + C_{cc}\right) \quad [\mathrm{F/m}] \qquad (2\text{-}52)$$

$$\hat{C}_M = \frac{1}{3}\left(C_{ab} + C_{bc} + C_{ca}\right) \quad [\mathrm{F/m}] \qquad (2\text{-}53)$$

$\hat{\boldsymbol{C}}_{\boldsymbol{P}}$ is a symmetrical matrix whose diagonal terms $\hat{C}_S$ are positive, and whose off-diagonal terms $\hat{C}_M$ are negative. This indicates when a positive line-to-neutral-voltage is applied to one phase, a positive charge is induced on that phase and negative charge are induced on the other phases, which is physically correct.

The shunt phase admittance matrix is given by:

$$\mathbf{Y_P} = j\omega \mathbf{C_P} = j(2\pi f)\mathbf{C_P} \quad [\mathrm{S/m}] \qquad (2\text{-}54)$$

or, for a completely transposed line:

$$\hat{\mathbf{Y}}_\mathbf{P} = j\omega \hat{\mathbf{C}}_\mathbf{P} = j(2\pi f)\hat{\mathbf{C}}_\mathbf{P} \quad [\mathrm{S/m}] \qquad (2\text{-}55)$$

Symmetrical component transformation of (2-55) will give:

$$\hat{\mathbf{Y}}_S = j(2\pi f)\begin{bmatrix} \hat{C}_0 & 0 & 0 \\ 0 & \hat{C}_1 & 0 \\ 0 & 0 & \hat{C}_2 \end{bmatrix} \tag{2-56}$$

where

$$\hat{C}_0 = \hat{C}_S + 2\hat{C}_M \quad [\text{F/m}] \tag{2-57}$$

$$\hat{C}_1 = \hat{C}_2 = \hat{C}_S - \hat{C}_M \quad [\text{F/m}] \tag{2-58}$$

Since $\hat{C}_M$ is negative, it can be concluded from equations (2-57) and (2-58) that the zero-sequence capacitance, $\hat{C}_0$, is usually much less than the positive- or negative-sequence capacitance.

The value calculated by (2-58) for positive- and negative-sequence capacitance will exceed the value given by (2-38) by only about 1% [B4]. Both the effect of the neutral conductors and the earth can therefore be neglected when calculating the positive- and negative-sequence capacitance. In the case of zero-sequence capacitance the neutral conductors may increase the capacitance by an amount of the order of 10%, so that its presence cannot be neglected as may be done for the positive and negative sequences.

2.2.3 AC transmission line models for short, medium and long lines

Consider the two-port network in Fig. 2.5, which is a generalised representation of an overhead line. The subscript S denotes sending-end quantities and the subscript R receiving-end quantities.

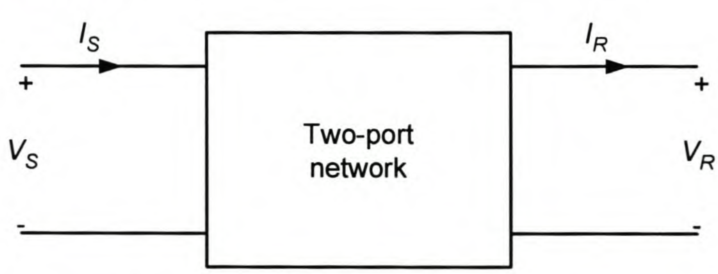


Fig. 2.5: Representation of a two-port network [B5]

The relation between sending- and receiving-end quantities may be expressed in the following general form:

$$V_S = AV_R + BI_R \quad [\mathrm{V}] \tag{2-59}$$

$$I_S = CV_R + DI_R \quad [\mathrm{A}] \tag{2-60}$$

or, in matrix format,

$$\begin{bmatrix} V_S \\ I_S \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} \tag{2-61}$$

The ABCD parameters, sometimes called *generalised circuit constants*, depend on the transmission-line parameters R, L, C and G. The ABCD parameters are, in general, complex numbers. A and D are dimensionless. B and C have the dimensions of ohms and siemens (mhos), respectively.

Short and medium line approximations

Fig. 2.6 represents a short line, usually applied to overhead lines less than 80 km long. Only the series resistance and reactance are included, whereas the shunt admittance is neglected. The circuit applies to either single-phase or completely transposed three-phase lines operating under balanced conditions [B5]. For a completely transposed three-phase line, Z is the positive-sequence series impedance, V_S and V_R the positive-sequence line-to-neutral voltages, and I_S and I_R the positive-sequence line currents. Similar zero-sequence and negative-sequence circuits can be employed to represent unbalanced operating conditions, using the parameters derived in the previous section.

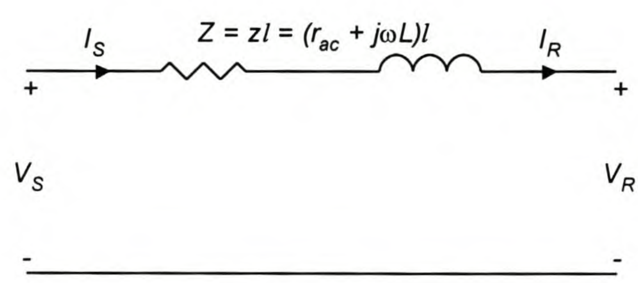


Fig. 2.6: Short line [B5]

From Fig. 2.6 and Fig. 2.7 the following notation is used:

Series impedance per unit length:

$$z = r_{ac} + j\omega L \quad [\Omega/\mathrm{km}] \tag{2-62}$$

Shunt admittance per unit length:

$$y = j\omega C \quad [\text{S/km}] \tag{2-63}$$

Total series impedance:

$$Z = zl \quad [\Omega] \tag{2-64}$$

Total shunt admittance:

$$Y = yl \quad [\text{S}] \tag{2-65}$$

where l is the length of the line in kilometers. The subscript 1, indicating positive sequence quantities, is omitted in this section.

Writing a KVL and KCL for the short line in Fig. 2.6:

$$V_S = V_R + ZI_R \tag{2-66}$$

$$I_S = I_R \tag{2-67}$$

or in matrix format

$$\begin{bmatrix} V_S \\ I_S \end{bmatrix} = \begin{bmatrix} 1 & Z \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} \tag{2-68}$$

Comparing equations (2-68) and (2-61), the ABCD parameters for the short line are:

$$A = D = 1 \quad [\text{per unit}] \tag{2-69}$$

$$B = Z \quad [\Omega] \tag{2-70}$$

$$C = 0 \quad [\text{S}] \tag{2-71}$$

For medium-length lines, ranging from 80 to 250 km, it is common to lump the total shunt capacitance and locate half at each end of the line [B5]. Such a circuit is called a *nominal π circuit*, shown in Fig. 2.7.

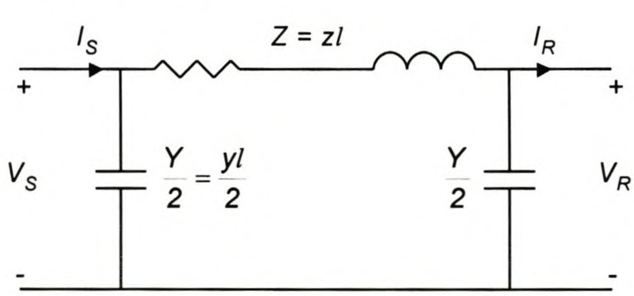


Fig. 2.7: Medium-length line - nominal π circuit [B5]

Writing a KVL and KCL for the medium-length line in Fig. 2.7 the following equation is obtained, written in matrix format as:

$$\begin{bmatrix} V_S \\ I_S \end{bmatrix} = \begin{bmatrix} \left(1+\frac{YZ}{2}\right) & Z \\ Y\left(1+\frac{YZ}{4}\right) & \left(1+\frac{YZ}{2}\right) \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} \tag{2-72}$$

Comparing equations (2-72) and (2-61), the ABCD parameters for the medium-length line are:

$$A = D = 1+\frac{YZ}{2} \quad [\text{per unit}] \tag{2-73}$$

$$B = Z \quad [\Omega] \tag{2-74}$$

$$C = Y\left(1+\frac{YZ}{4}\right) \quad [\text{S}] \tag{2-75}$$

Overhead-line differential equations

Long overhead-line equations (in terms of the ABCD parameters) will be discussed, as these represent an exact model for the overhead line. Fig. 2.1 represents a line section of length Δx and are repeated in Fig. 2.8 for convenience.

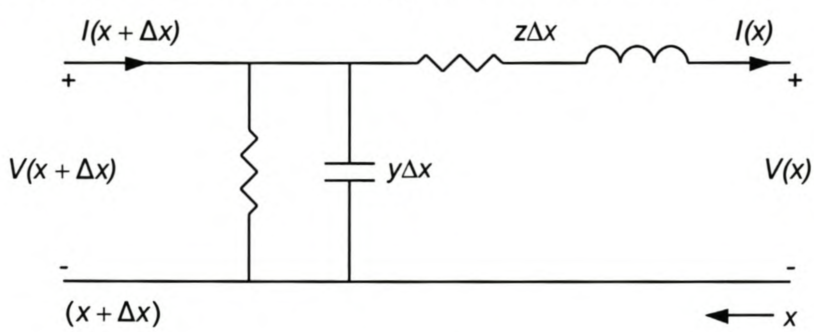


Fig. 2.8: Overhead line section of length Δx [B5]

The position x is measured from the receiving end of the line. The circuit constants:

$$z = r_{ac} + j\omega L \quad [\Omega/\text{km}] \tag{2-76}$$

$$y = G + j\omega C \quad [\text{S/km}] \tag{2-77}$$

are distributed uniformly throughout the length of the line. G is usually neglected for overhead lines [B7].

Equation (2-78) gives the ABCD parameters of the distributed line as a function of distance:

$$\begin{bmatrix} V(x) \\ I(x) \end{bmatrix} = \begin{bmatrix} A(x) & B(x) \\ C(x) & D(x) \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} \tag{2-78}$$

where

$$A(x) = D(x) = \cosh(\gamma x) \quad [\text{per unit}] \tag{2-79}$$

$$B(x) = Z_c \sinh(\gamma x) \quad [\Omega] \tag{2-80}$$

$$C(x) = \frac{1}{Z_c} \sinh(\gamma x) \quad [\text{S}] \tag{2-81}$$

$Z_c = \sqrt{z/y}$ $[\Omega]$ is called the *characteristic impedance* of the line, and $\gamma = \sqrt{zy}$ km^{-1} the *propagation constant.*

Equation (2-78) gives the voltage and current anywhere along the line, at a distance x from the receiving end. Letting $x = l$, the length of the line in kilometres, we obtain the voltage and current at the sending end of the line:

$$V_S = V_R \cosh(\gamma l) + I_R Z_c \sinh(\gamma l) \tag{2-82}$$

$$I_S = \frac{V_R}{Z_c} \sinh(\gamma l) + I_R \cosh(\gamma l) \tag{2-83}$$

The ABCD parameters given by equations (2-79) to (2-81) are exact parameters valid for any line length. For accurate calculations these equations must be used for overhead lines longer than 250 km [B5]. For quick calculations by hand, the approximate parameters discussed in the previous sections for short or medium lines can be used. For more accurate calculations the exact ABCD parameters must be used as well for short and medium lines.

Many computer programs, used in power system analysis and design, assume a circuit representation of an overhead line. The *overhead line equivalent π circuit*, shown in Fig. 2.9, are therefore a more convenient way of representing the terminal characteristics of an overhead line than the ABCD parameters of equations (2-79) to (2-81).

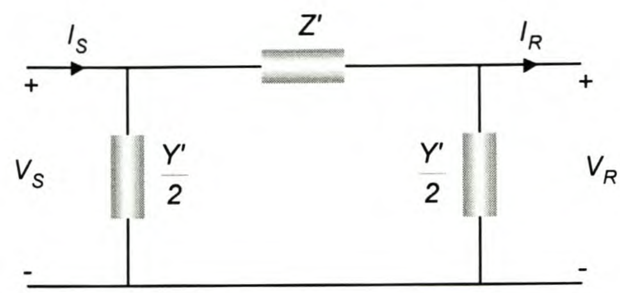


Fig. 2.9: Overhead line equivalent π circuit [B5]

From Fig. 2.9:

$$Z' = Z_c \sinh(\gamma l) \tag{2-84}$$

$$\frac{Y'}{2} = \frac{\tanh(\gamma l/2)}{Z_c} \tag{2-85}$$

Fig. 2.9 is identical in structure to the nominal π circuit of Fig. 2.7, except that Z' and Y' are used instead of Z and Y. Z' and Y' (equations (2-84) and (2-85)) are chosen in such a way that the equivalent π circuit will have the same ABCD parameters as those of the distributed line (equations (2-79) to (2-81)). Table 2-4 summarises the ABCD parameters for short, medium and long overhead lines.

Table 2-4: Summary of Overhead Line Parameters [B5]

Parameter	$A = D$ [per unit]	B [Ω]	C [S]
Short line (less than 80 km)	1	Z	0
Medium line – nominal π circuit (80 to 250 km)	$1+\frac{YZ}{2}$	Z	$Y\left(1+\frac{YZ}{4}\right)$
Long line – equivalent π circuit (more than 250 km)	$\cosh(\gamma l) = 1+\frac{Y'Z'}{2}$	$Z_c \sinh(\gamma l) = Z'$	$(1/Z_c)\sinh(\gamma l)$ $= Y'\left(1+\frac{Y'Z'}{4}\right)$

2.2.4 Implementation

This section focuses on the development of the Overhead Line Parameters (OLP) software tool, which was written as an extension to the input interface of PSAT (Power System Analysis Tool) to automatically calculate the equivalent impedances of a transmission line. OLP is restricted to calculating the series impedance and shunt admittance of balanced three-phase, three-wire lines, where it is assumed that the lines are perfectly transposed. Because currently only three-phase, balanced circuits are considered in PSAT, these restrictions are acceptable.

An ACSR conductor and tower database is implemented in OLP. Input data consist of selecting a phase conductor, a tower configuration and the number of phase conductors in a bundle. The values for bundle spacing, earth resistivity and distance of the line must also be entered. Output data consist of the series impedance and shunt admittance of different ACSR conductor sizes, bundle selections and tower configurations. The equivalent π circuit discussed in section 2.2.3 is also calculated. Provision is already made for unbalanced conditions as both the positive- and zero-sequence values are calculated. The results can be exported to either the sending or the receiving end in the input interface of PSAT [A1], [C2].

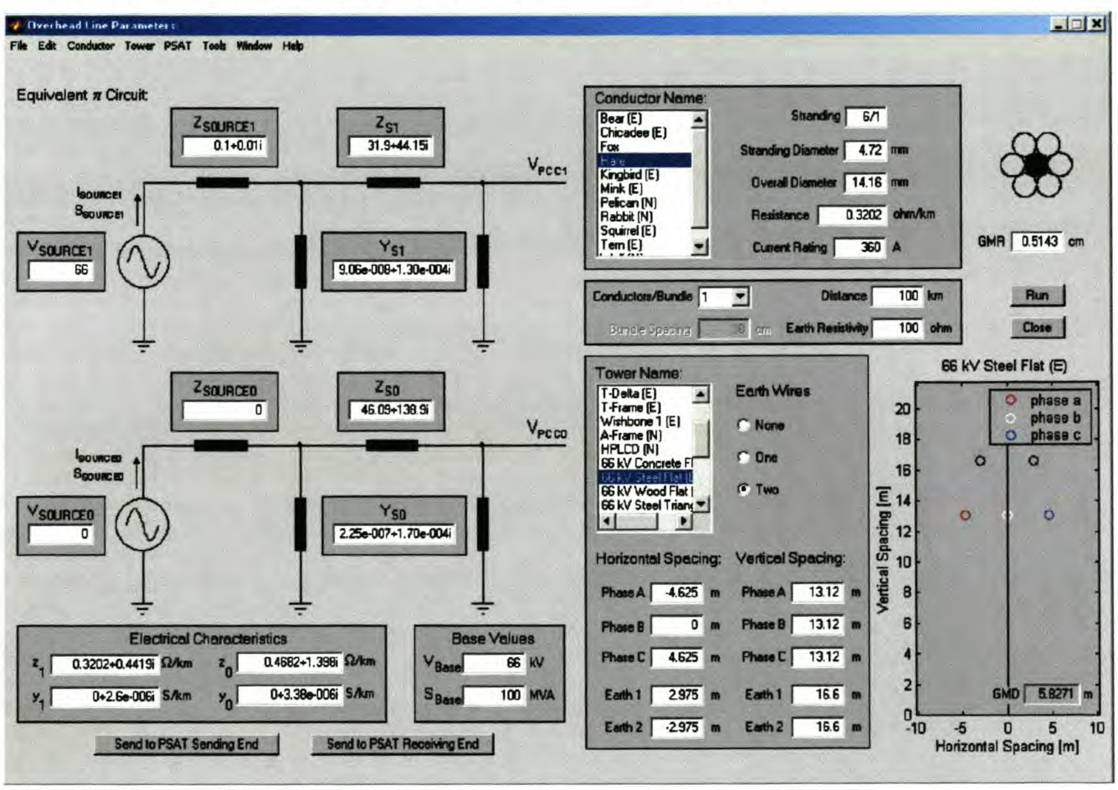
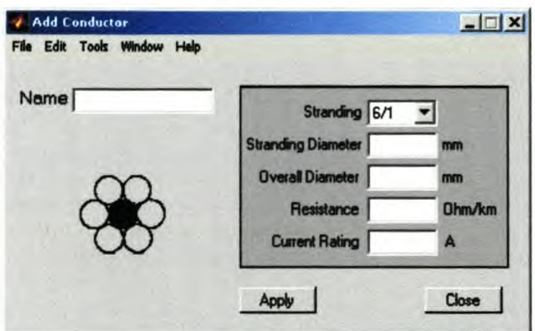


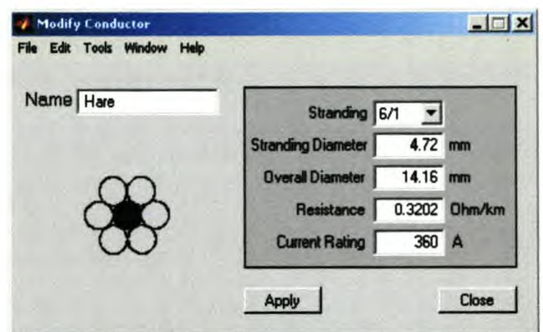
Fig. 2.10: Overhead line parameters interface

The OLP interface is shown in Fig. 2.10. Different conductors are selected from the conductor database by clicking on the conductor name in the *Conductor Name* listbox. Similarly, different towers can be selected from the tower database by clicking on the tower name in the *Tower Name* listbox. The tower structure is graphically portrayed as a single black line in the center: the red, blue and white circles represent the three phases and the black circles the neutral wires. The calculated sequence values for the series impedance and shunt admittance are displayed in the *Electrical Characteristics* box and are used to calculate the equivalent π circuit.

To add new conductors to the conductor database or modify existing conductors in the database, *Add Conductor* or *Modify Conductor* can be selected from the *Conductor* menu in the OLP interface. The add conductor and modify conductor interface of OLP is shown in Fig. 2.11.



(a)



(b)

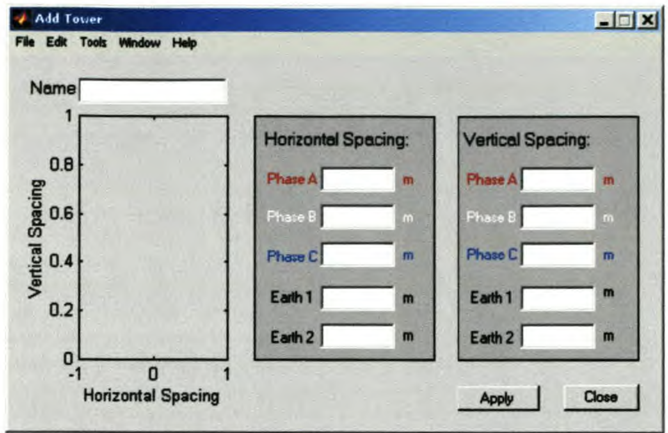
Fig. 2.11: a) Add conductor and b) Modify conductor interface

Similarly, new towers can be added to the tower database or existing towers can be modified by selecting *Add Tower* or *Modify Tower* from the *Tower* menu in the OLP interface. These interfaces are shown in Fig. 2.12.

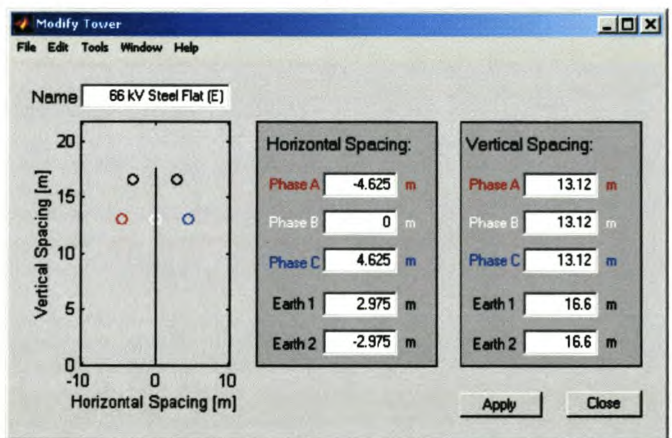
OLP is used to quickly calculate the series impedance, shunt admittance and equivalent π circuit for different ACSR conductor sizes, bundle selections and tower configurations. Calculation time and effort are therefore greatly reduced and the results can be exported to PSAT.

Software development

OLP is implemented using Matlab, developed by Mathworks Inc. Matlab was chosen as it is fundamentally a mathematical package, but it also supports the development of object-orientated graphical user interfaces (GUIs). Equations and formulae implemented in Matlab can easily be exported to other simulation packages that allow the user to create custom devices. It also provides excellent graphing tools that are not always available in other simulation packages.



(a)



(b)

Fig. 2.12: a) Add tower and b) Modify tower interface

Matlab GUIs are built around figures. A function is linked to each object in the figure and is executed according to the selections made or inputs given by the user. Matlab provides excellent printed and on-line documentation regarding GUIs [B12], and therefore the functions pertaining to the GUI and other graphics are not discussed here.

The equations to calculate the series impedance, shunt admittance and equivalent π circuit for a specific tower structure with different ACSR conductors, bundle selections, distances and earth resistivity are implemented in *conductor.m*. The equations to calculate the series impedance, shunt admittance and equivalent π circuit for different tower configurations are implemented in *tower.m*. The functions to add new conductors and modify existing conductors or to add new towers and modify existing towers are included in *conductor.m* and *tower.m* respectively. The development of the equations was discussed in sections 2.2.1 to 2.2.3. The full program listing for *conductor.m* and *tower.m* is included in Addendum B.1.

2.3 THE DC TRANSMISSION LINE MODEL

Fig. 2.13 represents a line section of length Δx. Whereas the configuration of AC lines is determined by the number of phases, the equivalent for DC lines are the poles. Since there is no inductance, there will not be any stability problems. There will be no line-charging currents, because there is no capacitance. Therefore, there will be no reactive power injected into or absorbed by the power system and only real power will flow.

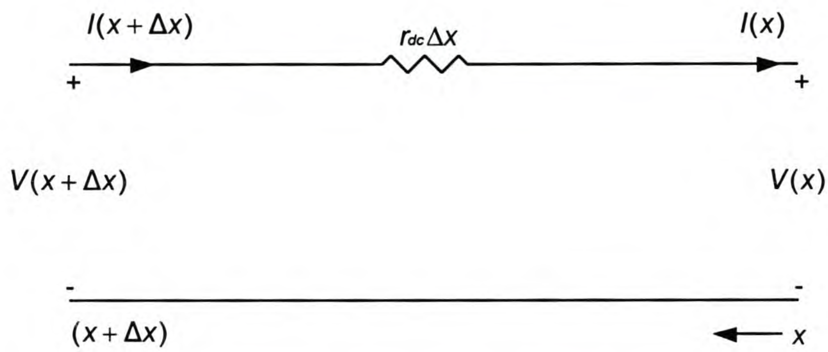


Fig. 2.13: DC line section of length Δx

The DC resistance of a conductor at a specified temperature T is:

$$r_{dc,T} = \frac{\rho_T l}{A} \quad [\Omega] \tag{2-86}$$

where ρ_T = conductor resistivity at temperature T

l = conductor length

A = conductor cross-sectional area.

For stranded conductors alternate layers of strands are spiralled in opposite directions to hold the strands together. This makes the strands 1 or 2% longer than the actual conductor length. As a result of this, the DC resistance of a stranded conductor is 1 or 2% larger than that calculated from equation (2-86) for a specified conductor length. It is very seldom that resistance is calculated from first principles, since the DC resistance values are readily available from the conductor tables at 20°C. These values also include the spiralling effect for stranded conductors. Equation (2-2) is used to correct r_{dc} to the required operating temperature as was discussed in detail in section 2.2.1.

For bundled conductors, the effective resistance per unit length of the bundled conductor is equal to r_{dc}/n, where n is the amount of conductors per bundle (n = 2, 3, or 4) [B5].

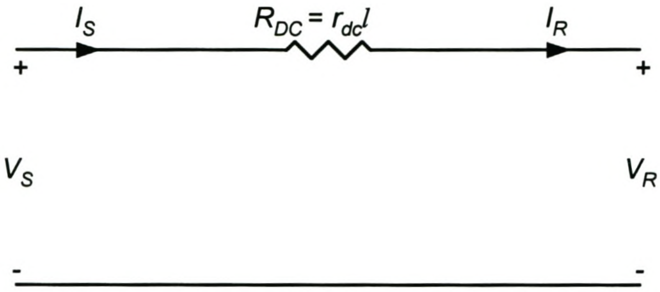


Fig. 2.14: DC transmission line model

The DC transmission line model is illustrated in Fig. 2.14. Writing a KVL and KCL for the line:

$$V_S = V_R + R_{DC} I_{DC} \tag{2-87}$$

$$I_S = I_R = I_{DC} \tag{2-88}$$

Comparing equations (2-167), (2-168) and (2-61) the ABCD parameters for the DC line are:

$$A = D = 1 \quad [\text{per unit}] \tag{2-89}$$

$$B = R_{DC} \quad [\Omega] \tag{2-90}$$

$$C = 0 \quad [\text{S}] \tag{2-91}$$

2.4 VOLTAGE REGULATION AND LINE LOSSES FOR AC TRANSMISSION LINES

Voltage regulation and efficiency (percentage losses) are fundamental properties of the performance of a transmission line. Losses depend on the impedance of a line. For AC transmission, line impedance is a function of the phasing (three-phase, phase-phase, etc.), conductor AC resistance and conductor spacing [C6].

Line losses can be reduced by either increasing the conductor size or by increasing the supply voltage for constant power loads or lowering it for constant impedance loads. Where the load is of constant power type, like most industrial loads, increasing the supply voltage is effective. The drawback of increasing the voltage is that the equipment is stressed, which in turn reduces the lifespan. This increases the probability of flashovers and failures, which in turn reduce the reliability of supply [C6]. Therefore losses cannot be reduced through acceptable means other than increasing the conductor size, which is a very costly option.

2.4.1 Power flow in terms of ABCD constants

Although power flow at any point along an overhead line can be found if the voltage, current and power factor at that point are known or can be calculated, equations for power can also be derived in terms of ABCD constants. Also, many times the only known variables in the system are the sending-end voltage at the source bus (also called the slack or swing bus), the line length, the line parameters and the receiving-end power.

The two-port network of Fig. 2.5 is repeated in Fig. 2.15 for convenience, as well as equations (2-59) and (2-60).

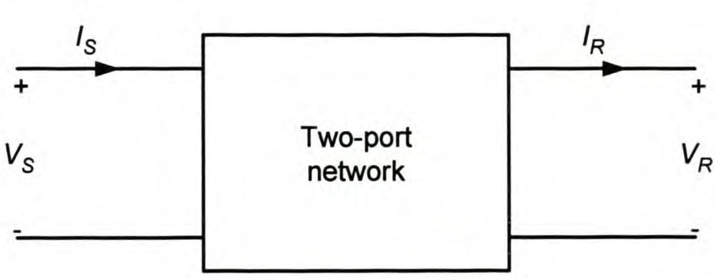


Fig. 2.15: Representation of a two-port network [B5]

From equations (2-59) and (2-60):

$$V_S = AV_R + BI_R \quad [\mathrm{V}] \tag{2-92}$$

$$I_S = CV_R + DI_R \quad [\mathrm{A}] \tag{2-93}$$

The following notation is used:

$$A = |A|\angle\alpha \tag{2-94}$$

$$B = |B|\angle\beta \tag{2-95}$$

$$V_R = |V_R|\angle 0° \tag{2-96}$$

$$V_S = |V_S|\angle\delta \tag{2-97}$$

The ABCD parameters are derived for line-to-neutral voltages; therefore V_S and V_R must also be treated as such.

Substituting (2-94) and (2-95) into (2-92) and solving for I_R:

$$I_R = \frac{1}{|B|}\left(|V_S|\angle(\delta-\beta) - |A||V_R|\angle(\alpha-\beta)\right). \tag{2-98}$$

Then, the complex power per phase $V_R I_R^*$ at the receiving end is:

$$S_R = P_R + jQ_R = \frac{|V_R|}{|B|}\left(|V_S|\angle(\beta-\delta) - |A||V_R|\angle(\beta-\alpha)\right) \tag{2-99}$$

where I_R^* is the complex conjugate of I_R.

The real power (P_R) and reactive power (Q_R) delivered to the receiving end are:

$$P_R = \mathrm{Re}(S_R) = \frac{|V_R||V_S|}{|B|}\cos(\beta-\delta) - \frac{|A||V_R|^2}{|B|}\cos(\beta-\alpha) \tag{2-100}$$

$$Q_R = \mathrm{Im}(S_R) = \frac{|V_R||V_S|}{|B|}\sin(\beta-\delta) - \frac{|A||V_R|^2}{|B|}\sin(\beta-\alpha) \tag{2-101}$$

2.4.2 Calculation of the sending-end voltage

The content of this section and sections 2.4.3 to 2.4.4 is based on work done by Smit [C7].

Only the magnitude $|V_S|$ of the sending-end voltage is usually known, where the angle δ is actually unknown and must be eliminated from equations (2-100) and (2-101). The next few steps indicate how to calculate V_S from the basic information available.

Solving (2-100) for $\cos(\beta - \delta)$:

$$\cos(\beta-\delta)=\frac{|B|}{|V_R||V_S|}\left(P_R+\frac{|A||V_R|^2}{|B|}\cos(\beta-\alpha)\right)=K_1 \tag{2-102}$$

and similarly (2-101) for $\sin(\beta - \delta)$:

$$\sin(\beta-\delta)=\frac{|B|}{|V_R||V_S|}\left(Q_R+\frac{|A||V_R|^2}{|B|}\sin(\beta-\alpha)\right)=K_2 \tag{2-103}$$

The angle δ is unknown and is eliminated from (2-102) and (2-103) using the following identity:

$$\cos^2(\beta-\delta)+\sin^2(\beta-\delta)=1 \Rightarrow K_1^2+K_2^2=1 \tag{2-104}$$

Substituting equations (2-102) and (2-103) into equation (2-104):

$$\begin{aligned}&\left(\frac{|B|}{|V_R||V_S|}\right)^2\left(P_R+\frac{|A||V_R|^2}{|B|}\cos(\beta-\alpha)\right)^2+\\&\left(\frac{|B|}{|V_R||V_S|}\right)^2\left(Q_R+\frac{|A||V_R|^2}{|B|}\sin(\beta-\alpha)\right)^2=1\end{aligned} \tag{2-105}$$

Solving for V_S from (2-105) in terms of V_R, P_R and Q_R:

$$|V_S|^2=\left(\frac{|B|}{|V_R|}\right)^2\left[\left(P_R+\frac{|A||V_R|^2}{|B|}\cos(\beta-\alpha)\right)^2+\left(Q_R+\frac{|A||V_R|^2}{|B|}\sin(\beta-\alpha)\right)^2\right] \tag{2-106}$$

From equation (2-106):

$$\left(P_R+\frac{|A||V_R|^2}{|B|}\cos(\beta-\alpha)\right)^2=P_R^2+2P_R\frac{|A||V_R|^2}{|B|}\cos(\beta-\alpha)+\frac{|A|^2|V_R|^4}{|B|^2}\cos^2(\beta-\alpha) \tag{2-107}$$

and

$$\left(Q_R+\frac{|A||V_R|^2}{|B|}\sin(\beta-\alpha)\right)^2=Q_R^2+2Q_R\frac{|A||V_R|^2}{|B|}\sin(\beta-\alpha)+\frac{|A|^2|V_R|^4}{|B|^2}\sin^2(\beta-\alpha) \tag{2-108}$$

With $S_R^2=P_R^2+Q_R^2$ and using the same identity as in (2-104), (2-106) can be simplified to:

$$|V_S|^2=\frac{|B|^2S_R^2}{|V_R|^2}+|A|^2|V_R|^2+2|A||B|[P_R\cos(\beta-\alpha)+Q_R\sin(\beta-\alpha)] \tag{2-109}$$

The angles α and β are network dependent and are eliminated by replacing the angles with the appropriate values of the equivalent π model of the line. It is also preferable to work with the real variables available, as they are easier to implement in a program or spreadsheet.

Using the equivalent π model of Fig. 2.9 and equations (2-84) and (2-85), the following formulae apply:

$$Z'=Z_C\sinh(\delta l)=R'+jX' \tag{2-110}$$

$$\frac{Y'}{2}=\frac{\tanh(\gamma l)}{Z_C}=\frac{G'}{2}+j\frac{B'}{2} \tag{2-111}$$

The shunt conductance G', introduced in (2-111), is small and its effect is often neglected [B5], [B11].

From Table 2-4 the following:

$$
\begin{aligned}
&A = |A|\angle\alpha = 1 + \frac{Y'Z'}{2} = D = |D|\angle\Delta \\
&\Rightarrow A = 1 + (R' + jX')\left(\frac{jB'}{2}\right) \\
&\Rightarrow A = \left(1 - \frac{B'X'}{2}\right) + j\frac{R'B'}{2}
\end{aligned}
\tag{2-112}
$$

and

$$
\begin{aligned}
&B = |B|\angle\beta = Z' \\
&\Rightarrow B = R' + jX'
\end{aligned}
\tag{2-113}
$$

In the equations to follow the subscript (′) indicating equivalent π circuit values will be neglected for the sake of simplicity and to avoid confusion. From equations (2-112) and (2-113) it further follows that:

$$
\cos\alpha = \cos\Delta = \frac{\left(1 - \frac{BX}{2}\right)}{|D|} \tag{2-114}
$$

$$
\sin\alpha = \sin\Delta = \frac{RB}{2|D|} \tag{2-115}
$$

$$
\cos\beta = \frac{R}{|Z|} \tag{2-116}
$$

and

$$
\sin\beta = \frac{X}{|Z|} \tag{2-117}
$$

Substituting the values for $|A|$ and $|B|$ into equation (2-109) and remembering that $|B|$ is one of the transmission line parameters equal to Z' with the subscript neglected:

$$
|V_S|^2 = \frac{|Z|^2 S_R^2}{|V_R|^2} + |D|^2|V_R|^2 + 2|D||Z|\left[P_R\cos(\beta - \alpha) + Q_R\sin(\beta - \alpha)\right] \tag{2-118}
$$

Using the following identities:

$$
\cos(\beta - \alpha) = \cos\beta\cos\alpha + \sin\beta\sin\alpha \tag{2-119}
$$

$$
\sin(\beta - \alpha) = \sin\beta\cos\alpha - \cos\beta\sin\alpha \tag{2-120}
$$

equation (2-118) can now be rewritten as:

$$|V_S|^2 = \frac{|Z|^2 S_R^2}{|V_R|^2} + |D|^2|V_R|^2 + 2|D||Z|\begin{bmatrix} P_R(\cos\beta\cos\alpha + \sin\beta\sin\alpha) \\ + Q_R(\sin\beta\cos\alpha - \cos\beta\sin\alpha) \end{bmatrix} \tag{2-121}$$

By further substitution of equations (2-114) to (2-117) into equation (2-121):

$$\begin{aligned} |V_S|^2 &= \frac{|Z|^2 S_R^2}{|V_R|^2} + |D|^2|V_R|^2 \\ &+ 2|D||Z|\left[P_R\left(\frac{R}{|Z|}\frac{\left(1-\frac{BX}{2}\right)}{|D|} + \frac{X}{|Z|}\frac{RB}{2|D|}\right) + Q_R\left(\frac{X}{|Z|}\frac{\left(1-\frac{BX}{2}\right)}{|D|} - \frac{R}{|Z|}\frac{RB}{2|D|}\right)\right] \end{aligned} \tag{2-122}$$

Substituting $|D|^2$ in equation (2-122) with:

$$|D|^2 = \left(1 - \frac{BX}{2}\right)^2 + \left(\frac{RB}{2}\right)^2 \tag{2-123}$$

and simplifying

$$|V_S|^2 = \frac{|Z|^2 S_R^2}{|V_R|^2} + \left[1 - BX + \left(\frac{B|Z|}{2}\right)^2\right]|V_R|^2 + 2RP_R + Q_R\left(2X - B|Z|^2\right) \tag{2-124}$$

where

$$|Z|^2 = R^2 + X^2 \tag{2-125}$$

2.4.3 Calculation of the receiving-end voltage

If $|V_S|$ is known $|V_R|$ can be calculated by evaluating equation (2-124) with the general expression for a quadratic equation:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{2-126}$$

where x is the unknown quantity and a, b and c are numerical factors referred to as coefficients of the equation.

Equation (2-124) can be rewritten as:

$$|V_S|^2 - 2RP_R - Q_R\left(2X - B|Z|^2\right) = \frac{|Z|^2 S_R^2}{|V_R|^2} + \left[1 - BX + \left(\frac{B|Z|}{2}\right)^2\right]|V_R|^2 \tag{2-127}$$

Equation (2-127) is reduced to:

$$F = \frac{|Z|^2 S_R^2}{|V_R|^2} + K|V_R|^2 \tag{2-128}$$

where

$$F = |V_S|^2 - 2RP_R - Q_R(2X - B|Z|^2) \tag{2-129}$$

and

$$K = 1 - BX + \frac{B^2|Z|^2}{4} \tag{2-130}$$

Equation (2-128) is now written in the form of (2-126) as:

$$|V_R| = \sqrt{\frac{F + \sqrt{\left(F^2 - 4|Z|^2 S_R^2 K\right)}}{2K}} \tag{2-131}$$

The receiving-end voltage ($|V_R|$) can now be determined by calculating the values for F and K in (2-129) and (2-130) and substituting these values into (2-131).

After $|V_R|$ is determined from equation (2-131), the *voltage regulation* of the line can be expressed as:

$$V_d = (V_S - V_R) \times 100\% \tag{2-132}$$

where V_S and V_R are per unit quantities.

2.4.4 Calculation of the sending-end power

In order to calculate the line losses the sending-end power needs to be calculated. The complex power at the sending end of the line is given by:

$$S_S = V_S I_S^* \tag{2-133}$$

From section 2.4.1 the sending-end voltage and current was expressed in terms of the ABCD parameters as:

$$V_S = AV_R + BI_R \quad [\mathrm{V}] \tag{2-134}$$

$$I_S = CV_R + DI_R \quad [\mathrm{A}] \tag{2-135}$$

I_R is the only unknown variable in (2-135) and can be solved for by substituting (2-134) into (2-135) and simplifying:

$$
\begin{aligned}
I_S &= CV_R + \frac{D}{B}(V_S - AV_R) \\
\Rightarrow I_S &= \frac{D}{B}V_S - \frac{V_R}{B}(AD - BC) \\
\Rightarrow I_S &= \frac{1}{B}(DV_S - V_R)
\end{aligned}
\tag{2-136}
$$

where $AD - BC = 1$.

The complex conjugate of I_S will thus be:

$$
I_S^* = \frac{1}{|B|}\left(|D||V_S|\angle(\beta - \Delta - \delta) - |V_R|\angle\beta\right) \tag{2-137}
$$

The complex power at the sending end is calculated by substituting (2-137) into (2-133) and simplifying:

$$
S_S = P_S + jQ_S = \frac{|V_S|}{|B|}\left(|D||V_S|\angle(\beta - \Delta) - |V_R|\angle(\beta + \delta)\right) \tag{2-138}
$$

The real power (P_S) and reactive power (Q_S) at the sending end are:

$$
P_S = \frac{|D||V_S|^2}{|B|}\cos(\beta - \Delta) - \frac{|V_S||V_R|}{|B|}\cos(\beta + \delta) \tag{2-139}
$$

$$
Q_S = \frac{|D||V_S|^2}{|B|}\sin(\beta - \Delta) - \frac{|V_S||V_R|}{|B|}\sin(\beta + \delta). \tag{2-140}
$$

The angle δ in equations (2-139) and (2-140) are unknown and must be eliminated, using equations (2-102) and (2-103), which are repeated in (2-141) and (2-142) for convenience.

$$
\cos(\beta - \delta) = K_1 \tag{2-141}
$$

$$
\sin(\beta - \delta) = K_2 \tag{2-142}
$$

Using trigonometric identities (2-141) and (2-142) can be rewritten as:

$$\cos(\beta-\delta)=\cos\beta\cos\delta+\sin\beta\sin\delta=K_1$$
$$\Rightarrow \cos(\delta)=\frac{(K_1-\sin\beta\sin\delta)}{\cos\beta} \tag{2-143}$$

and

$$\sin(\beta-\delta)=\sin\beta\cos\delta-\cos\beta\sin\delta=K_2$$
$$\Rightarrow \sin\delta=\frac{(\sin\beta\cos\delta-K_2)}{\cos\beta} \tag{2-144}$$

Substitute (2-143) into (2-144) and simplify:

$$\sin\delta=\frac{\left(\sin\beta.\frac{(K_1-\sin\beta.\sin\delta)}{\cos\beta}-K_2\right)}{\cos\beta}$$
$$\Rightarrow \sin\delta\left(1+\frac{\sin^2\beta}{\cos^2\beta}\right)=\frac{\sin\beta}{\cos^2\beta}K_1-\frac{K_2}{\cos\beta} \tag{2-145}$$
$$\Rightarrow \sin\delta=\sin(\beta)K_1-\cos(\beta)K_2$$

Equation (2-145) is now substituted back into (2-143) and simplified:

$$\cos\delta=\frac{(K_1-\sin\beta[\sin(\beta)K_1-\cos(\beta)K_2])}{\cos\beta}$$
$$\Rightarrow \cos\delta=\frac{K_1}{\cos\beta}(1-\sin^2\beta)+\sin(\beta)K_2 \tag{2-146}$$
$$\Rightarrow \cos\delta=\cos(\beta)K_1+\sin(\beta)K_2$$

Expanding equation (2-139):

$$P_S=\frac{|D||V_S|^2}{|B|}\cos(\beta-\Delta)-\frac{|V_S||V_R|}{|B|}(\cos\beta\cos\delta-\sin\beta\sin\delta) \tag{2-147}$$

The term $(\cos\beta\cos\delta-\sin\beta\sin\delta)$ can now be rewritten as:

$$\cos\beta\cos\delta-\sin\beta\sin\delta=K_1(\cos^2\beta-\sin^2\beta)+K_2(\cos\beta\sin\beta+\cos\beta\sin\beta)$$
$$\Rightarrow K=K_1(\cos^2\beta-\sin^2\beta)+2K_2\cos\beta\sin\beta \tag{2-148}$$

The values of K_1 and K_2 (equations (2-102) and (2-103)) are now substituted into (2-148):

$$K = \frac{|B|\left(\cos^2\beta - \sin^2\beta\right)}{|V_R||V_S|}\left(P_R + \frac{|A||V_R|^2}{|B|}\cos(\beta-\alpha)\right) + \frac{2\cos\beta\sin\beta|B|}{|V_R||V_S|}\left(Q_R + \frac{|A||V_R|^2}{|B|}\sin(\beta-\alpha)\right) \tag{2-149}$$

Substitute (2-149) into (2-147) and simplify:

$$P_S = \frac{|D||V_S|^2}{|B|}\cos(\beta-\Delta) - \left(\cos^2\beta - \sin^2\beta\right)\left(P_R + \frac{|A||V_R|^2}{|B|}\cos(\beta-\alpha)\right) - 2\cos\beta\sin\beta\left(Q_R + \frac{|A||V_R|^2}{|B|}\sin(\beta-\alpha)\right) \tag{2-150}$$

$$\Rightarrow P_S = \frac{|D||V_S|^2}{|B|}\cos(\beta-\Delta) - M - N$$

where

$$M = \left(\cos^2\beta - \sin^2\beta\right)\left(P_R + \frac{|A||V_R|^2}{|B|}\cos(\beta-\alpha)\right) \tag{2-151}$$

and

$$N = 2\cos\beta\sin\beta\left(Q_R + \frac{|A||V_R|^2}{|B|}\sin(\beta-\alpha)\right). \tag{2-152}$$

The angles α, β, and Δ are network dependent and are replaced by the appropriate values of the equivalent π circuit model. From section 2.4.2 the following:

$$\cos\alpha = \cos\Delta = \frac{\left(1 - \frac{BX}{2}\right)}{|D|} \tag{2-153}$$

$$\sin\alpha = \sin\Delta = \frac{RB}{2|D|} \tag{2-154}$$

$$\cos\beta = \frac{R}{|Z|} \tag{2-155}$$

$$\sin\beta = \frac{X}{|Z|} \tag{2-156}$$

Expanding the sine and cosine terms in equations (2-150) - (2-152) and substituting equations (2-153) - (2-156):

$$P_S = \frac{|D||V_S|^2}{|Z|}\left(\frac{R}{|Z|}\frac{\left(1-\frac{BX}{2}\right)}{|D|}+\frac{X}{|Z|}\frac{RB}{2|D|}\right)-M-N \tag{2-157}$$

with

$$M = \left(\frac{R^2-X^2}{|Z|^2}\right)\left[P_R+\frac{|A||V_R|^2}{|Z|}\left(\frac{R}{|Z|}\frac{\left(1-\frac{BX}{2}\right)}{|D|}+\frac{X}{|Z|}\frac{RB}{2|D|}\right)\right] \tag{2-158}$$

and

$$N = 2\frac{RX}{|Z|^2}\left[Q_R+\frac{|A||V_R|^2}{|Z|}\left(\frac{X}{|Z|}\frac{\left(1-\frac{BX}{2}\right)}{|D|}+\frac{R}{|Z|}\frac{RB}{2|D|}\right)\right]. \tag{2-159}$$

Substitute (2-158) and (2-159) into (2-157) and simplify:

$$\begin{aligned}P_S &= \frac{|V_S|^2 R - P_R\left(R^2-X^2\right)-2RXQ_R}{|Z|^2}\\ &\quad -\frac{|V_R|^2}{|Z|^4}\left(R\left(R^2-X^2\right)+2RX\left(X-\frac{B}{2}|Z|^2\right)\right)\end{aligned} \tag{2-160}$$

Some of the terms in (2-160) are now reduced as follows:

$$\begin{aligned}&\left(R^2-X^2\right)=R^2-X^2+2R^2-2R^2\\ \Rightarrow &\left(R^2-X^2\right)=2R^2-|Z|^2\end{aligned} \tag{2-161}$$

and

$$R\left(R^2-X^2\right)+2RX\left(X-\frac{B}{2}|Z|^2\right)=R|Z|^2(1-BX). \tag{2-162}$$

Substituting (2-161) and (2-162) into (2-160) and simplifying:

$$P_S = \frac{R}{|Z|^2}\left(\left(|V_S|^2-|V_R|^2\right)-2\left(RP_R+XQ_R\right)+BX|V_R|^2\right)+P_R \tag{2-163}$$

The real line losses can be defined as the difference between the sending-end real power P_S and the receiving-end real power P_R:

$$P_{line\ loss} = \frac{R}{|Z|^2}\left(\left(|V_S|^2 - |V_R|^2\right) - 2\left(RP_R + XQ_R\right) + BX|V_R|^2\right) \tag{2-164}$$

The receiving-end reactive power Q_R can be calculated in the same way as the real power P_R:

$$Q_S = \frac{X}{|Z|^2}\left(|V_S|^2 - |V_R|^2 - 2\left(RP_R + XQ_R\right)\right) - \frac{B}{|Z|^2}\left(\frac{|Z|^2}{2}\left(|V_S|^2 - |V_R|^2\right) + R^2|V_R|^2\right) \tag{2-165}$$

Similarly the reactive line losses can be defined as the difference between the sending-end reactive power Q_S and the receiving-end reactive power Q_R:

$$Q_{line\ loss} = \frac{X}{|Z|^2}\left(|V_S|^2 - |V_R|^2 - 2\left(RP_R + XQ_R\right)\right) - \frac{B}{|Z|^2}\left(\frac{|Z|^2}{2}\left(|V_S|^2 - |V_R|^2\right) + R^2|V_R|^2\right) \tag{2-166}$$

2.5 VOLTAGE REGULATION AND LINE LOSSES FOR DC TRANSMISSION LINES

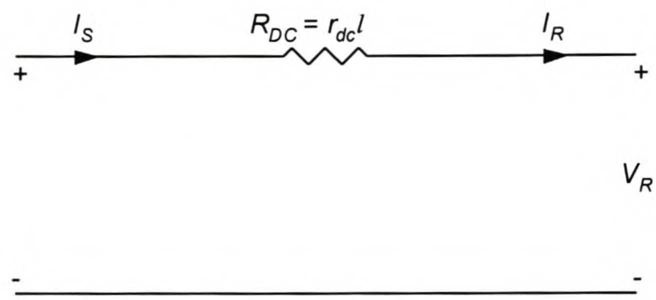


Fig. 2.16: DC transmission line model

The DC transmission line model is illustrated in Fig. 2.14 and is repeated in Fig. 2.16 for the purposes of this discussion. Writing a KVL and KCL for the line:

$$V_S = V_R + R_{DC} I_{DC} \quad [\text{V}] \tag{2-167}$$

$$I_S = I_R = I_{DC} \quad [\text{A}] \tag{2-168}$$

Comparing equations (2-167), (2-168) and (2-61) the ABCD parameters for the DC line are:

$$A = D = 1 \quad [\text{per unit}] \tag{2-169}$$

$$B = R_{DC} \quad [\Omega] \tag{2-170}$$

$$C = 0 \quad [\text{S}] \tag{2-171}$$

In the case of DC transmission line inductance and capacitance have no effect and equations (2-169) to (2-171) will be treated as real values, where the angles are ignored. The ABCD parameters are derived for line-to-neutral voltages; therefore V_S and V_R must also be treated as such.

The real power per pole at the receiving end is:

$$P_R = V_R I_R \tag{2-172}$$

Rewriting (2-172) in terms of I_R and substituting into (2-167):

$$V_S = R_{DC} \frac{P_R}{V_R} + V_R \tag{2-173}$$

which can also be solved for $|V_R|$.

Rewriting (2-173) in terms of (2-126) with $|V_R|$ as the unknown variable:

$$V_R = \frac{V_S + \sqrt{\left(V_S^2 - 4R_{DC}P_R\right)}}{2} \tag{2-174}$$

Losses for DC transmission are dependent on the number of poles (monopolar or bipolar) and the conductor DC resistance. Line losses can be defined as the difference between the sending-end power P_S and the receiving-end power P_R:

$$P_{line\ loss} = P_S - P_R \tag{2-175}$$

The sending-end power is unknown and must be calculated:

$$P_S = V_S I_S \tag{2-176}$$

Substituting (2-168), (2-172), (2-176) into (2-175) and simplifying:

$$\begin{aligned} P_{line\ loss} &= V_S I_S - V_R I_R \\ \Rightarrow P_{line\ loss} &= I_{DC}\left(V_S - V_R\right) \\ \Rightarrow P_{line\ loss} &= P_R\left(\frac{V_S}{V_R} - 1\right) \end{aligned} \tag{2-177}$$

2.6 TRANSMISSION COSTS

Whenever there are different investment schemes with either different technologies or different scales for an engineering project, a technical and economic quantitative analysis should be carried out and an economic assessment should be given in order to select the optimal scheme [B13]. The economic analysis is done in terms of the *present value of money* (PVM). Present value of money is a method of comparing costs and savings that occur at different times on a consistent and equitable basis for decision-making [B14]. Inflation is not taken into account in the transmission cost equations since it has equal influence on savings and costs.

When comparing transmission costs for an AC or DC interconnection, it is necessary to evaluate not only the investment cost of the line and terminals, but also the cost of losses in order to select the most suitable scheme. Converter losses are another important parameter that must be taken into consideration when calculating DC transmission costs [A7].

In this section the procedure to calculate the transmission costs for both AC and DC technologies will be discussed. Transmissions costs are also referred to in literature as the cost of capital and losses or LCC. The term LCC will be adopted for the purposes of this discussion.

2.6.1 Review of the LCC

a) The cost of losses per annum, $CL(n)$, can be calculated as:

$$CL(n) = 8760 \cdot ELF \cdot GC \cdot P_{line\ loss} \quad [R] \tag{2-178}$$

where ELF = Energy loss factor

GC = Generation cost [R/kWh]

Shown in Fig. 2.17 is $CL(n)$, which is directly proportional to the square of the annual load factor (ALF) of the line or feeder. ALF is the actual energy supplied over a period divided by the maximum demand over that period multiplied by the time period selected (i.e. actual energy supplied divided by potential energy supplied) [B14]. This proportionality can be quantified by the ELF [C8] as:

$$ELF = x \cdot ALF + (1-x) \cdot ALF^2 \tag{2-179}$$

where $x = 0.2$ (determined by statistical evaluation).

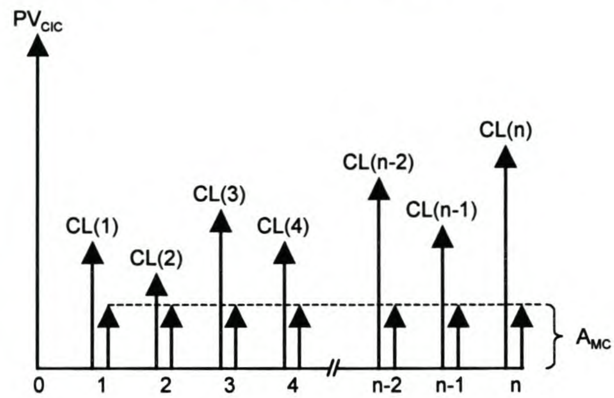


Fig. 2.17: A graphical representation of the relationship between PV_{CIC}, A_{MC} and $CL(n)$

The generation cost, also referred to as the de-escalated energy cost [c/kWh], is fixed for each year and was obtained from the financial department of Eskom. $P_{line\ loss}$ was determined for AC and DC lines in sections 2.4.4 and 2.5 respectively.

Networks losses (P_{Losses}) include the losses in both the line technologies and any additional network support technologies. The cost of compensator losses must be included in the transmission costs of AC networks requiring compensation. Subsequently, converter losses must be taken into account for DC transmission costs. The cost of line and support technology losses can be calculated as follows:

$$\begin{aligned} CL(n) &= 8760 \cdot ELF \cdot GC \cdot \left(P_{line\ loss} + P_{support\ loss}\right) \\ \Rightarrow CL(n) &= 8760 \cdot ELF \cdot GC \cdot P_{Losses} \quad \text{[R/yr]} \end{aligned} \tag{2-180}$$

where $P_{support\ loss}$ = kW losses of the support technologies

P_{Losses} = Total kW losses

The present value of the annual cost of losses (PV_{CL}) can be calculated from the interest rate (i) as:

$$PV_{CL}(n) = CL(n)\frac{1}{(1+i)^n} \tag{2-181}$$

where n is the number of years.

The interest term (i) was introduced in (2-181). In the economic assessment of an engineering project the real meaning of interest is different in both concept and value from what is involved in an investment in a bank. It is the result of profit in the process of the use of capital. In order to differentiate between the two concepts, the term discount rate is substituted for interest rate [B13]. The term discount factor (*DF*) can, therefore, be defined from equation (2-181) as:

$$DF(n) = \frac{1}{(1+i)^n} \tag{2-182}$$

The present value of the total cost of losses (PV_{TCL}) can now be calculated over the entire life cycle of the line. This is done be summating all the PV_{CL} terms calculated in (2-181):

$$\begin{aligned} PV_{TCL} &= \sum_{n=1}^{years} PV_{CL}(n) \\ \Rightarrow PV_{TCL} &= \sum_{n=1}^{years} CL(n) \cdot DF(n) \end{aligned} \tag{2-183}$$

b) The determination of the annual equivalent maintenance cost (A_{MC}) for the line is regularly ignored when doing an economic assessment of AC or DC line technologies. This is because the maintenance cost of one conductor is the same as for another conductor. It would therefore have no significant effect on the reduction of the LCC [C9]. However, when AC and DC line options are economically weighted against each other, A_{MC} must be considered because the maintenance cost for AC and DC technologies differ. Fig. 2.17 shows the A_{MC} over the lifetime of the line. The total maintenance cost over the life-cycle of the line can be calculated in terms of the present value of money as:

$$PV_{TMC} = A_{MC} \frac{(1+i)^n - 1}{i(1+i)^n} \tag{2-184}$$

A_{MC} for conventional AC line technologies (no compensation required) will include only the annual maintenance cost of the line:

$$A_{MC} = A_{MC(line)} \tag{2-185}$$

A_{MC} for DC line technologies or AC line technologies with compensation will include the annual maintenance cost of the line and the support technology:

$$A_{MC} = A_{MC(line)} + A_{MC(support)} \tag{2-186}$$

c) The capital investment cost (CIC) is the initial cost of the line and includes the following [C9]:

- Tower structure;
- Foundations;
- Conductor;
- Ground wire;
- Insulation
- Hardware
- Tower erection;
- Stringing;
- Extra work.

The capital investment cost of the two converter stations must be included in the PV_{CIC} (present value of the CIC) for the DC line technologies. For AC line technologies requiring compensation, the capital investment cost of the support technologies is included. The relationship between PV_{CIC}, A_{TMC} and $CL(n)$ is shown in Fig. 2.17.

d) The present value of life-cycle cost (PV_{LCC}) is calculated from the initial capital investment (PV_{CIC}), an annual maintenance component (PV_{TMC}) and the cost of losses (PV_{TCL}):

$$PV_{LCC} = PV_{CIC} + PV_{TMC} + PV_{TCL} \tag{2-187}$$

2.6.2 Determination of the break-even distance

If the transmission distance is great enough, the saving in capital cost and lower losses with DC transmission line may be enough to pay for the two converter stations, one at either end [A8]. This is known as the *break-even distance*. According to Hingorani, the AC vs DC issue in simplistic, generalised terms of break-even distance amounts to [A9]:

- What is the difference in cost between AC and DC lines, taking into account the construction cost and the cost of losses? This difference will favor DC.
- What is the difference in cost between AC substations (including reactive power compensation) and the DC converter stations? This difference will favour AC.

An interesting point is that, as the power requirement is decreased, the saving from AC to DC transmission line per MW-km would increase faster than the increase in the cost of converters, and therefore the break-even distance would generally decrease [A9]. An example is used in the next paragraph to illustrate how the break-even distance is determined.

Consider a 400 kV HVAC line for 2000 MW at a cost of R800 000 per kilometer (or R400 per MW-km). The converter costs of the alternative DC solution are about R1 000 per kW. Allowing R400 per kW for the AC substations, a total of R1200 million must be saved from the line cost to break even. If an HVDC line cost R240 per MW-km, it would take 3750 kilometers of 2000 MW HVDC to break even with AC.

2.7 THE APPROPRIATE TECHNOLOGY INDEX (ATI) FOR AC AND DC LINE TECHNOLOGIES

The ATI is used to provide a quantative measure of the best technical and economic option. The results are not absolute, due to the influence of weighting factors, and are project specific. Key performance indicators (KPIs) are summated using weighting factors to obtain the overall technical and economical performance of a line technology [C9].

The ATI is defined as:

$$ATI = K_1 \cdot w_1 + K_2 \cdot w_2 + K_3 \cdot w_3 + K_4 \cdot w_4 \tag{2-188}$$

where w_1 to w_4 are weighting factors quantifying the importance of the KPIs, K_1 to K_4, for a specific project. The KPIs are determined by the technology applied whereas the weighting factors are project specific.

The philosophy of determining the ATI of various line technologies has been successfully applied in the past for HV AC lines at various voltage levels. Recently it has also been applied for MV AC and LV AC lines [C6]. Where compensation is part of the AC line technology, its effect has been included in the various KPIs. The results of these studies are discussed in section 2.7.1.

2.7.1 ATI for AC line technologies

Fig. 2.18 provides a summary of the KPIs for AC line technologies at the three major voltage levels (HV, MV and LV). At HV levels the KPIs to consider are life-cycle costs, capacity, stability and voltage regulation. MV and LV networks rarely suffer from stability problems and this KPI is therefore ignored for MV and LV levels. Similarly the ATI technique can be applied to LV networks with an added KPI in the form of measuring the dynamic response of the voltage regulator employed to compensate for voltage fluctuations due to sudden load changes.

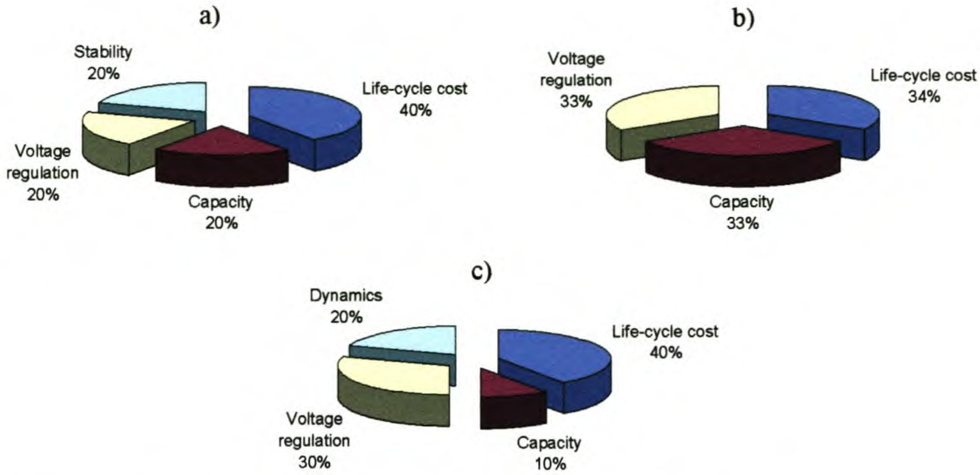


Fig. 2.18: KPIs for AC line technologies at various voltage levels: a) HV b) MV and c) LV [C6]

The first KPI, K_1, is the total LCC normally taken over a period of 25 years. The calculations of this KPI for conventional AC networks and for compensated AC networks are discussed in detail in section 2.6.1, and only the formula used to represent the total LCC in terms of the PVM will be repeated here:

$$K_1 = PV_{LCC} = PV_{CIC} + PV_{TMC} + PV_{TCL} \quad \text{[R]} \tag{2-189}$$

The second KPI, K_2, is the line cost per probabilistic power rating of the line technology to be compared. This KPI is an indication of the importance of the continuity of supply under heavy-loaded conditions above its previous deterministic thermal current rating [C6]. For conventional AC line technologies:

$$K_2 = \frac{PV_{CI}}{MVA_{emergency}} = \frac{PV_{CIC}}{\sqrt{3} \cdot U \cdot I \cdot N \cdot l} \quad \text{[R/km/MVA]} \tag{2-190}$$

For AC lines with compensation, the additional capital cost of the support technology must be added to P_{CIC}. With additional compensation it is also possible to increase the power transfer capacity of the power network. For compensated AC lines:

$$\begin{aligned} K_2 &= \frac{PV_{CI(line)} + PV_{CI(support)}}{MVA_{emergency}} = \frac{PV_{CIC(line)} + PV_{CIC(support)}}{\sqrt{3} \cdot U \cdot I \cdot N \cdot l + \Delta S} \\ \Rightarrow K_2 &= \frac{PV_{CI}}{MVA_{emergency}} = \frac{PV_{CIC}}{\sqrt{3} \cdot U \cdot I \cdot N \cdot l + \Delta S} \quad \text{[R/km/MVA]} \end{aligned} \tag{2-191}$$

where:

PV_{CI} = present value of the total capital investment (CI) [R/km]

$PV_{CI(line)}$ = present value of the CI of the line [R/km]

$PV_{CI(support)}$	=	present value of the CI of the compensator [R/km]
PV_{CIC}	=	present value of the total CIC [R]
$PV_{CIC(line)}$	=	present value of the CIC of the power line [R]
$PV_{CIC(support)}$	=	present value of the CIC of the compensator [R]
U	=	system voltage (line-line) [kV]
I	=	ampacity current at 80°C [A]
N	=	number of conductors in a bundle
l	=	line length [km]
$MVA_{emergency}$	=	emergency loading of the line
ΔS	=	apparent power increase due to compensation [MVA]

The *ampacity*, I, of a conductor can be defined as the current which will meet the design, security and safety criteria of a particular line on which the conductor is used [A11]. The ampacity can be determined by deterministic or probabilistic means using a number of formulae. A discussion on the deterministic and probabilistic ampacity ratings of conductors is presented in section 3.2.

The third variable, K_3, is the line cost per surge impedance loading (SIL) of the line technologies to be compared. This variable determines the quality of supply with a stability focus. Some HV lines are not suitable for loading above their SIL limit, because of problems associated with stability due to the fact that the network is mechanically controlled [C6]. This KPI is ignored for MV and LV line technologies, since stability is not a major problem on these networks.

$$K_3 = \frac{PV_{CI}}{MVA_{SIL}} = \frac{PV_{CIC} \cdot Z_e}{U^2 \cdot l} = \frac{PV_{CIC} \cdot \sqrt{\frac{|Z|}{Y}}}{U^2 \cdot l} \quad \text{[R/km/MVA]} \tag{2-192}$$

where

PV_{CI}	=	present value of the total capital investment [R/km]
PV_{CIC}	=	present value of the total capital investment cost [R]
Z_e	=	surge impedance of the line [Ω]
$\|Z\|$	=	magnitude of the positive sequence impedance [Ω]
Y	=	positive sequence [S]
U	=	system voltage (line-line) [kV]
l	=	line length [km]

MVA_{SIL} = surge impedance loading of the line

The fourth and last KPI, K_4, is the percentage voltage drop over the total length of the line. This measures the ability of the particular line technology to meet the set voltage regulation requirements [C6]. Voltage regulation were discussed in great detail in sections 2.4.2 and 2.4.3 and only the equations to calculate the receiving-end voltage (V_R) and the voltage regulation (V_d) will be repeated here:

$$|V_R| = \sqrt{\frac{F + \sqrt{\left(F^2 - 4|Z|^2 S_R^2 K\right)}}{2K}} \tag{2-193}$$

where

$$F = |V_S|^2 - 2RP_R - Q_R(2X - B|Z|^2) \tag{2-194}$$

and

$$K = 1 - BX + \frac{B^2|Z|^2}{4}. \tag{2-195}$$

In (2-194) and (2-195) R is the total line resistance in per unit, X the total line inductance in per unit and B the total line susceptance in per unit. P_R and Q_R are respectively the per unit real and reactive power at the receiving-end of the line. Also from (2-193) to (2-195) the following:

$$|Z|^2 = R^2 + X^2 \quad [\Omega] \tag{2-196}$$

From equation (2-132) the voltage regulation of the line can be expressed as:

$$V_d = (V_S - V_R) \times 100\% \tag{2-197}$$

where V_S and V_R are per unit quantities.

2.7.2 ATI for DC line technologies

Fig. 2.19 shows the KPIs for DC line technologies at the three major voltage levels (HV, MV and LV).

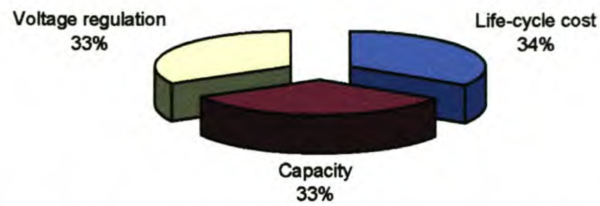


Fig. 2.19: KPIs for DC line technologies at HV, MV and LV levels

With DC no stability problems occur, because the AC systems are decoupled and the power can be freely and rapidly adjusted by converter control. The KPI (K_3) that determines the quality of supply with a stability focus or measures the dynamic response of the converter to compensate for voltage fluctuations at the load end can, therefore, be ignored for DC line technologies. Only the LCC, capacity and voltage regulation KPIs will be considered for DC line technologies.

The calculation of K_1 for DC line technologies are discussed in detail in section 2.6.1 and only the formula used to represent the total LCC in terms of the PVM will be repeated here:

$$K_1 = PV_{LCC} = PV_{CIC} + PV_{TMC} + PV_{TCL} \quad [\mathrm{R}] \qquad (2\text{-}198)$$

The line cost per probabilistic power rating of the DC line technologies to be compared (K_2) is derived in similar fashion as its AC counterpart. This KPI is an indication of the importance of the continuity of supply under heavy-loaded conditions. However, overloading is much more restricted in DC transmission, because the silicon-controlled rectifier has a very small thermal capacity and thus the modern converter valve is only designed to handle temporary over-currents. If long-term overload capacity is desired, appropriate overrating of the silicon devices and permitting higher temperatures at the converter plant can achieve this [B2]. For DC line technologies K_2 is defined as:

$$\begin{aligned} K_2 &= \frac{PV_{CI(line)} + PV_{CI(support)}}{MW_{emergency}} = \frac{PV_{CIC(line)} + PV_{CIC(support)}}{2 \cdot U \cdot I \cdot N \cdot l} \\ \Rightarrow K_2 &= \frac{PV_{CI}}{MW_{emergency}} = \frac{PV_{CIC}}{2 \cdot U \cdot I \cdot N \cdot l} \quad [\mathrm{R/km/MW}] \end{aligned} \qquad (2\text{-}199)$$

where:

PV_{CI}	=	present value of the total capital investment (CI) [R/km]
$PV_{CI(line)}$	=	present value of the CI of the line [R/km]

$PV_{CI(support)}$	=	present value of the CI of the converter [R/km]
PV_{CIC}	=	present value of the total CIC [R]
$PV_{CIC(line)}$	=	present value of the CIC of the power line [R]
$PV_{CIC(support)}$	=	present value of the CIC of the converter [R]
U	=	system voltage (pole-neutral) [kV]
I	=	ampacity current at 80°C [A]
N	=	number of conductors in a bundle
l	=	line length [km]
$MW_{emergency}$	=	emergency loading of the line

The voltage regulation KPI (K_4) was discussed in great detail in section 2.5 and only the equations to calculate the receiving-end voltage (V_R) and the voltage regulation (V_d) will be repeated here:

$$V_R = \frac{V_S + \sqrt{\left(V_S^2 - 4R_{DC}P_R\right)}}{2} \tag{2-200}$$

where R_{DC} = total DC line resistance in per unit

P_R = receiving-end real power in per unit

From equation (2-132) the voltage regulation of the line can be expressed as:

$$V_d = \left(V_S - V_R\right) \times 100\% \tag{2-201}$$

where V_S and V_R are per unit quantities.

2.7.3 Implementation of the ATI

In order to equalise the four independent KPIs to one measuring scale, the upper and lower limits per KPI have to be set. The upper limit represents the best-case result (score of 10) and a median value of 3 represents the average of existing solutions. The lower limit represents a score of zero. The rest of the values for each KPI are then linearly extrapolated between the high and low values. This is best illustrated with an example:

Example:

For K_1 the upper limit would be the line technology with the lowest LCC, say R1 000 000, and the lower limit the line technology with the highest LCC, say R5 000 000. The median value can then be calculated as follows:

$$Median\,Value = Lower\ Limit - \left(\frac{Lower\ Limit - Upper\ Limit}{10}\right)\cdot 3$$

$$\Rightarrow Median\,Value = R5\,000\,000 - \left(\frac{R5\,000\,000 - R1\,000\,000}{10}\right)\cdot 3$$

$$\Rightarrow Median\,Value = R3\,800\,000$$

A value of R3 000 000 is then linearly extrapolated between the low and high values as follows:

$$K_{1(extrapolated)} = 10 - \left(\frac{K_1 - Upper\ Limit}{Lower\ Limit - Upper\ Limit}\right)\cdot 10$$

$$\Rightarrow K_{1(extrapolated)} = 10 - \left(\frac{R3\,000\,000 - R1\,000\,000}{R5\,000\,000 - R1\,000\,000}\right)\cdot 10$$

$$\Rightarrow K_{1(extrapolated)} = 5$$

The values above are graphically portrayed in Fig. 2.20.

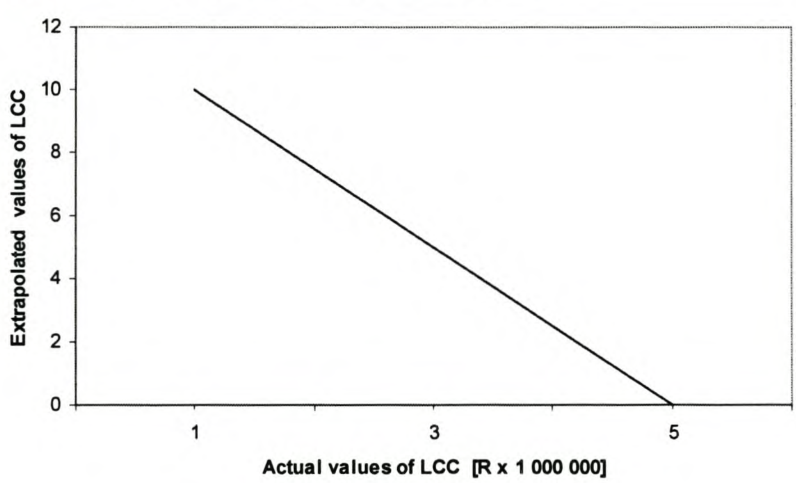


Fig. 2.20: Ranking for LCC evaluation

The weighting factors are used to indicate the dominance of a KPI. The values of the weighing factors are between 0 and 1 (1 = dominant and 0 = no influence at all). If, for example, the LCC is more important than the voltage drop of a line, then w_1 would be higher than w_4. The sum of all the weighting factors should be equal to one.

For HV AC line technologies the weighting factors (0.7 0.05 0.05 0.2) mean that the LCC (importance of initial capital expenditure and losses) is 70% dominant, the probabilistic current rating (capacity) and SIL (stability) are 5% dominant and the voltage drop is 20% dominant. The weighting factors are project specific and can be scaled to compare the sensitivity of each factor per chosen line technology. At present the following sensitivities can be evaluated:

- (1,0,0,0) – Normal operation
- (0,1,0,0) – Emergency operation only
- (0,0,1,0) – Surge impedance at HV or dynamic performance at LV levels
- (0,0,0,1) – Voltage drop only

2.8 IMPLEMENTATION

TESAT is used to evaluate the voltage regulation, line losses, LCC and ATIs of different AC and DC line technologies. The equations for voltage regulation, line losses, LCC and ATIs discussed in sections 2.4 to 2.7 are implemented in TESAT. These equations were derived using ABCD parameters and trigonometric identities to eliminate the voltage angle (δ), which is usually not known.

The AC interface of TESAT is shown in Fig. 2.21 and the DC interface in Fig. 2.22. The series impedance, shunt conductance, cost and ampacity rating (at 80°C) for three different conductors are specified in the *Compare* frame of the interfaces. Load data, system voltage and line length are also specified, as well as the interest rate, the support technology cost and the ALF. The values of the weighting factors ($w_1 - w_4$) are specified in the *ATI* frame of the interfaces.

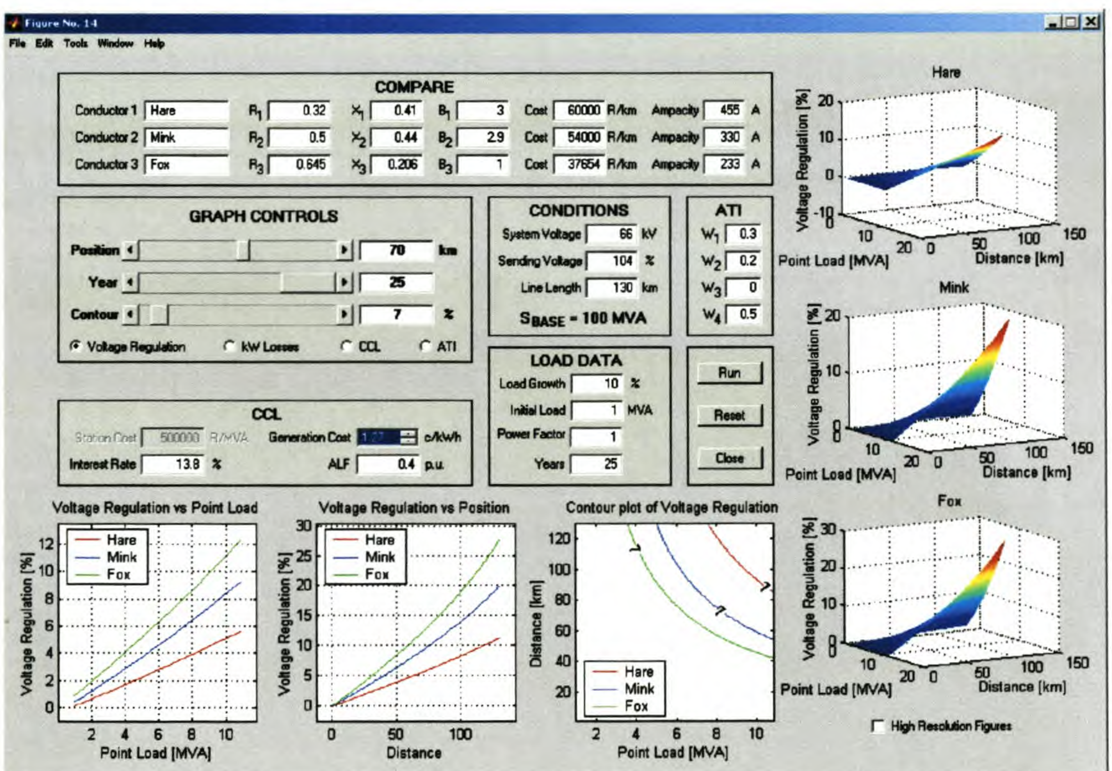


Fig. 2.21: AC interface of TESAT

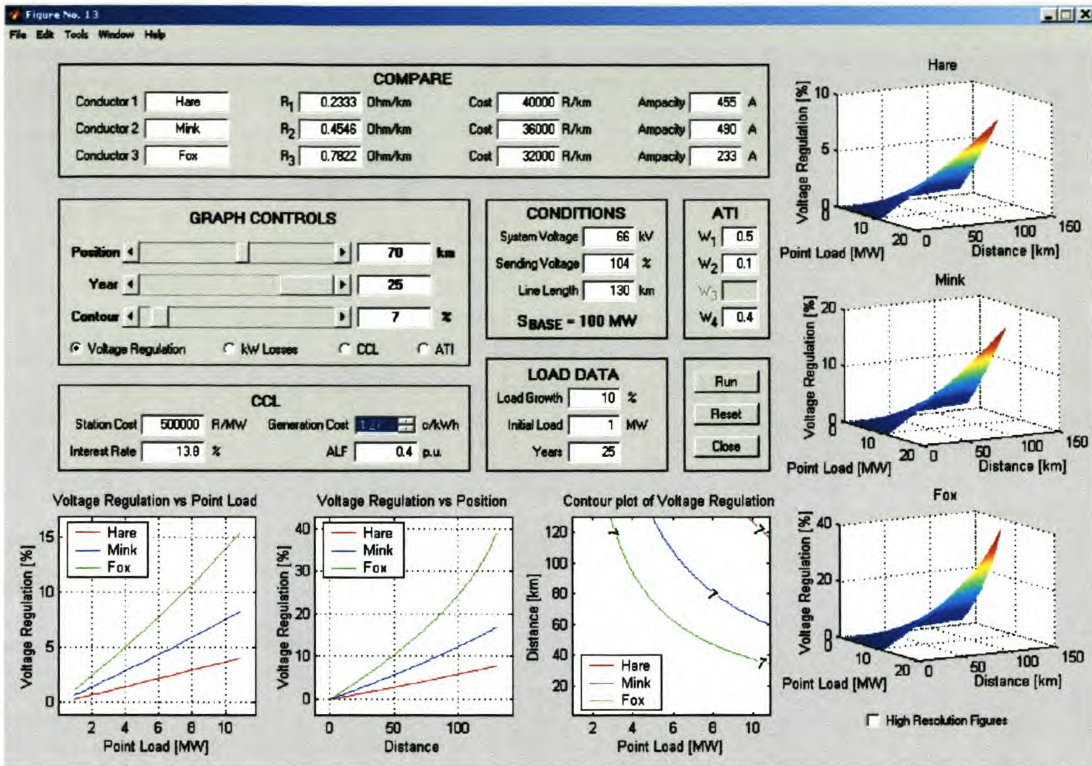


Fig. 2.22: DC interface of TESAT

In Fig. 2.21 and Fig. 2.22 three-dimensional graphs are used to portray the resulting voltage regulation, losses, LCC and ATI for the different AC and DC conductors respectively. The growth in point load is plotted on the X-axis and the line length on the Y-axis. For voltage regulation, losses, LCC and ATIs the magnitude is plotted on the Z-axis. In addition to the three-dimensional graphs three two-dimensional graphs are displayed. These graphs provide cross-sections through the three-dimensional graphs at user-selected points. One cross-section is parallel to the XZ-plane; one is parallel to the YZ-plane and the other parallel to the XY-plane (contour plot). These are labelled as *Voltage Regulation vs. Point Load*, *Voltage Regulation vs. Position* and *Contour plot of Voltage Regulation*, respectively. The user specifies the specific position, the specific year and the specific contour under *Graph Controls*. The parameters of interest (*Voltage regulation*, *kW losses*, *LCC* and *ATI*) are also selectable under *Graph Controls*. Fig. 2.21 and Fig. 2.22 displays the voltage regulation graphs, but it is easy to switch to losses, LCC or ATI graphs.

Software development

TESAT is implemented, just like OLP, in Matlab. The reasons for choosing Matlab and the use of Matlab GUIs were discussed in section 2.2.4.

The equations to calculate the voltage regulation, line losses, LCC and ATIs for the AC and DC technologies are implemented in *lfac.m* and *lfdc.m* respectively. These programs start by importing all the conductor parameters, system parameters, load parameters and financial parameters. The programs will only function properly if the user enters all the basic parameters before analysis. The same method is followed by *lfac.m* and *lfdc.m* for comparing different line conductors:

1. Calculate the receiving-end voltage and the voltage regulation for the different line technologies from the conductor parameters, system parameters and load parameters specified by the user.
2. With the receiving-end voltage known, the line losses can be calculated for the different lines.
3. With the line losses known, the LCC cost for each line technology can be calculated from the financial parameters.
4. With the LCC costs known, the rest of the ATIs for the different lines can be calculated from the financial parameters.

All the solutions will be real. Line technologies with complex solutions are highlighted in red to indicate that the specific conductor cannot be used. The solutions to the voltage regulation, line losses, LCC and ATIs for each technology are assigned to data matrixes that are used by the GUI to plot the various graphs. The full program listing for *lfac.m* and *lfdc.m* is included in Addendum B4.

TSAT uses basic information to quickly evaluate the voltage regulation, losses, LCC and ATIs of different conductors from an AC and DC perspective in order to select the optimum conductor for the task at hand. TESAT therefore greatly reduces the time and effort involved in the tedious calculations for voltage regulation, losses, LCC and ATIs.

CHAPTER 3 LOAD REACH AND TRANSFER CAPABILITIES OF AC AND DC LINE TECHNOLOGIES

3.1 INTRODUCTION

Much work has been done recently on the applicability of new line technologies in the stretching of distribution networks. This work included the identification of practical application opportunities for the Electronic Voltage Regulator (EVR) on MV and LV AC distribution networks [A10]. However, the recent advances in the semiconductor technology have also extended the economic power range for DC transmission to just a few MW. With DC transmission the load reach limits of these networks become something of the past. The aim of this chapter is to compare the load reach and transfer capabilities of various AC and DC line technologies as far as bulk energy transfer is concerned.

A discussion on the deterministic and probabilistic current ratings of conductors is presented in section 3.2. In 3.3 the thermal load reach of different AC and DC line technologies are compared at distribution voltage levels: firstly for the Nampower choice of conductors and then for the Eskom choice. It is also shown how deterministic and probabilistic current ratings affect the load reach of these conductors. The transfer capabilities for different AC and DC line technologies, distribution voltages and conductor sizes are presented in section 3.4.

3.2 THERMAL LOAD REACH – A DETERMINISTIC AND PROBABILISTIC APPROACH

Willis defines *load reach* as the distance an electrical line can move power before encountering limits in its performance due to engineering criteria [B14]. If the maximum allowable *voltage drop* across a line is set at 7.5% and a line has a voltage drop of 2.5% per kilometer at the recommended loading level, it follows that the line has a load reach of three kilometers or less. If a point load placed at the end of the line requires power to be moved three kilometers or less, then the line is capable of meeting these needs at the recommended loading. However, if the requirement is for power to be moved more than three kilometers, than either the line must be upgraded to one with less voltage drop per kilometer, or the size of the point load must be reduced.

Willis defines thermal load reach *as the distance a particular line type can move power at its thermal loading limit* [B14]. For many years the conductor thermal rating or ampacity has been determined by a deterministic method using conservative ambient conditions. It is a quick and simple method and has been used by Eskom and other utilities around the world. The deterministic approach assumes certain bad cooling conditions (low wind speed, high ambient temperature, etc.) and calculates the current that would result in the line design temperature being reached. The line templating or design temperature is that temperature at which the height of the conductor above the ground is the minimum permissible [A11]. The drawback of this method is that it does not address the safety or the relationship between safety and the power transfer capability.

However, in today's economic environment it is necessary to use assets more effectively and, on power lines, costs can be deferred or saved by finding ways to operate lines closer to their thermal limits. A new approach to that of weighing up the probability of risk of flashover due to environmental conditions and ground clearances has resulted in the so-called probabilistic ampacity ratings of conductors. The probabilistic approach provides the means to calculate the thermal rating at different templating temperatures. It also quantifies the probability of an unsafe condition arising associated with the rating and keeps this constant for conductors of a similar type. The probabilities applied are based on present probabilities so that if the line is utilised at a higher temperature, the probability of an unsafe condition arising is no more than at present.

The deterministic approach will be used for the Nampower choice of conductors and the probabilistic approach for Eskom conductor choices. The reason for this is firstly to illustrate the use of both the deterministic and probabilistic method to establish the table of conductor ampacities. Secondly, the probabilistic values at different template temperatures for some of the Nampower conductor choices are not readily available.

The ampacity values for the Nampower conductor selection at distribution voltage levels are shown in Table 3-1. The conductors in Table 3-1 are thermally rated for a 75°C electrical rating, but the line template temperature is at 50°C.

Table 3-1: Conductor ampacity table using deterministic loading techniques [C12]

Conductor Type	Templating Temperature [°C]	Normal Loading (75°C)		Emergency Loading (90°C)	
		$I = I_{AC}$ [A]	$I = I_{DC}$ [A]	$I = I_{AC}$ [A]	$I = I_{DC}$ [A]
Gopher	50	122	128	147	154
Fox	50	150	157	181	190
Hare	50	285	297	349	363
Rabbit	50	186	196	225	238
Oak	50	288	288	370	370

Table 3-2 shows the probabilistic ampacity values for the conductors used by Eskom at 11, 22 and 33 kV. For the purpose of generating the table the probability of an unsafe condition occurring was calculated for the Eskom design philosophy prior to 1987 [C11]. With this philosophy the conductor was thermally rated for a 75°C electrical rating, but the line template temperature was at 50°C. In the equations the probability of an unsafe condition was then kept constant and the ratings at different template temperatures were calculated.

Table 3-2: Conductor ampacity table using probabilistic loading techniques [C13], [C12]

Conductor Type	Templating Temperature [°C]	Normal Loading		Emergency Loading	
		$I = I_{AC}$ [A]	$I = I_{DC}$ [A]	$I = I_{AC}$ [A]	$I = I_{DC}$ [A]
Magpie	50	33	35	40	42
Magpie	60	47	49	52	55
Magpie	70	58	61	62	65
Magpie	80	67	70	70	73
Magpie	90	74	78	77	81
Magpie	100	81	85	84	88
Squirrel	50	106	111	135	142
Squirrel	60	130	136	160	168
Squirrel	70	149	156	181	190
Squirrel	80	165	173	198	208
Squirrel	90	178	187	213	223
Squirrel	100	190	199	227	238
Fox	50	148	155	192	201
Fox	60	184	193	228	239
Fox	70	210	220	258	271
Fox	80	233	244	283	297
Fox	90	253	265	305	320
Fox	100	270	283	324	340
Mink	50	209	220	272	286
Mink	60	258	271	324	341
Mink	70	297	313	367	386
Mink	80	330	347	402	423
Mink	90	357	376	434	457
Mink	100	382	402	461	485
Hare	50	292	304	380	396
Hare	60	357	372	454	473
Hare	70	408	425	515	536
Hare	80	455	474	565	588
Hare	90	496	516	609	634
Hare	100	529	551	685	713

It is important to note that if the values in Table 3-2 are to be used, it is of vital importance that the template temperature and the conductor thermal rating are the same, e.g. for a 50°C template temperature the normal and emergency values at 50°C electrical rating must be used. If the template temperature and conductor thermal rating are different the probability of an unsafe condition will not be the same as that for the calculated values in Table 3-2.

The conductor types and sizes specified in Table 3-1 and Table 3-2 have already been standardised by both Nampower and Eskom at distribution level and are adopted without change to this work [C14],[C15]. The conductor properties were sourced from the data sheets of overhead aluminium conductors and are included in Addendum A.1.

3.3 THERMAL LOAD REACH OF AC AND DC LINE TECHNOLOGIES AT DISTRIBUTION VOLTAGE LEVELS

The receiving-end voltage for three-phase AC lines was calculated in Chapter 2 as a function of the sending-end voltage, the receiving-end power and the impedance. From equation (2-127) the receiving-end voltage can be written as:

$$|V_S|^2 - 2r_{ac}lP_R - Q_R\left(2xl - bl\left(r_{ac}{}^2l^2 + x^2l^2\right)\right) = \frac{\left(r_{ac}^2l^2 + x^2l^2\right)S_R^2}{|V_R|^2} + \left[1 - blxl + \frac{b^2l^2\left(r_{ac}^2l^2 + x^2l^2\right)}{4}\right]|V_R|^2 \qquad (3\text{-}1)$$

where r_{ac} and x is the series impedance per kilometer and b the shunt admittance per kilometer.

The thermal load reach can be calculated by rewriting equation (3-1) in terms of the line length. The solution to the line length is a fourth-order solution. Explicit solutions exist for fourth-order equations, but the formulae are very long and tedious to work with. Maple, a symbolic mathematical software package from Waterloo, was used to solve (3-1) in terms of the line length l:

$$\sqrt{c_1l^4 + c_2l^3 + c_3l^2 + c_4l + c_5} = 0 \qquad (3\text{-}2)$$

where

$$c_1 = V_R^4b^2\left(r^2 + x^2\right) \qquad (3\text{-}3)$$

$$c_2 = -4Q_RV_R^2b\left(r^2 - b^2\right) \qquad (3\text{-}4)$$

$$c_3 = 4\left(S_R^2r^2 + S_R^2x^2 - V_R^4bx\right) \qquad (3\text{-}5)$$

$$c_4 = 8V_R^2\left(rP_R + xQ_R\right) \qquad (3\text{-}6)$$

$$c_5 = 4V_R^2\left(V_R^2 - V_S^2\right) \qquad (3\text{-}7)$$

Matlab was used to calculate the values for the constants $c_1 - c_5$ and the line length l. Addendum B5 contains the listing of *constants.m*, the file written to solve (3-2). Four possible solutions exist for the line length, and the correct solution must be chosen from the four possibilities. Two of the solutions to (3-2) are complex numbers and the other two are real numbers, of which one is a negative number. Both the complex solutions and the negative solution are ignored, as line length cannot be a complex or negative number.

Similarly the thermal load reach can be calculated for bipolar DC lines by rewriting equation (2-173) in terms of the line length:

$$l = \frac{V_R(V_S - V_R)}{r_{dc}P_R} \tag{3-8}$$

where r_{dc} is the dc resistance per kilometer.

Fig. 3.1 shows the deterministic thermal load reach of 33 kV three-phase AC and bipolar DC networks for various Nampower conductor types and sizes. Fig. 3.2 shows the probabilistic load reach of these networks for different Eskom conductors at a template temperature of 50°C. A power factor of 0.95 was assumed for the AC networks.

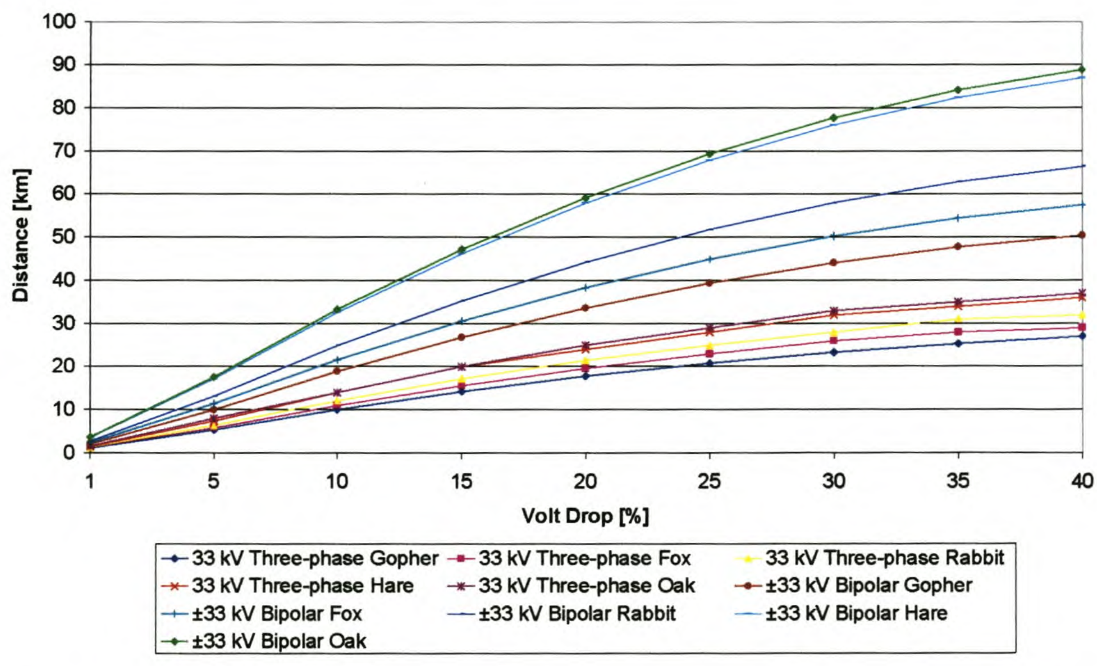


Fig. 3.1: Deterministic thermal load reach of different Nampower conductors from an AC and DC perspective

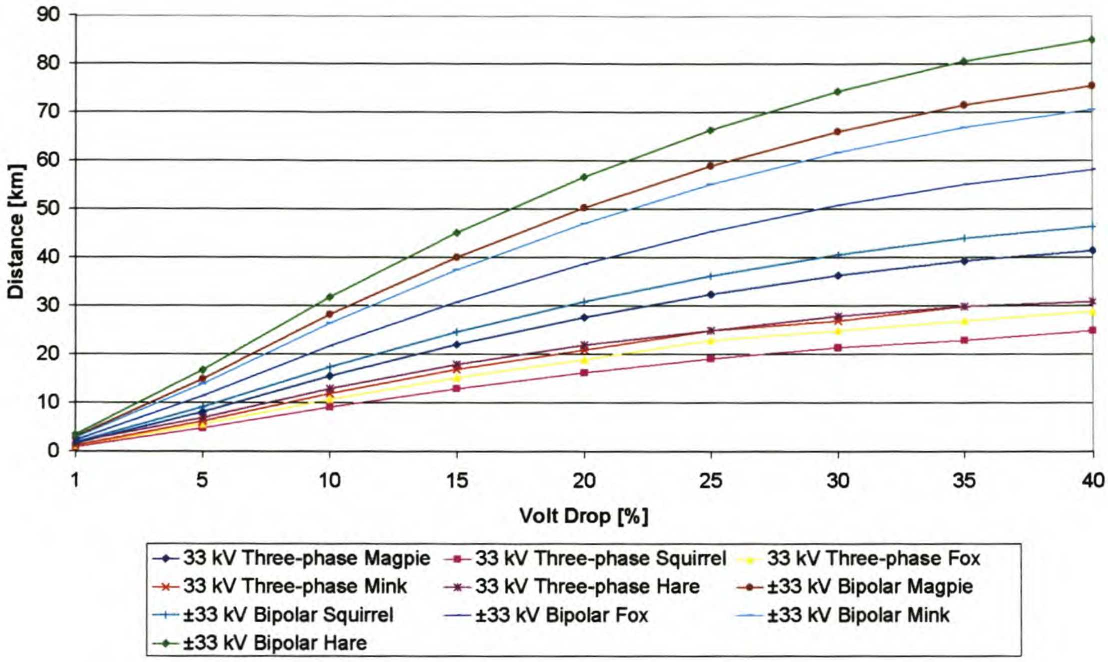


Fig. 3.2: Probabilistic thermal load reach of different Eskom conductors from an AC and DC perspective

Fig. 3.1 and Fig. 3.2 show that a reduction in voltage-drop restrictions implies an increased conductor length by more than three to four times. Cost can be significantly reduced on medium and low voltage AC networks by increasing the voltage regulation margin by more than 30% [A10]. Cost comparisons on different options to compensate for these low voltages were done. The study not only investigated the applicability of new line technologies in the stretching of AC distribution networks, but also innovative techniques for network support to establish longer load-reach capabilities. The cost savings were evaluated against the cost of boosting the reduced voltage levels to the required margins [A12].

The voltage drop limitations of AC networks become something of the past when applying DC line technologies. The voltage source converter at the load-end is able to compensate for a voltage drop of more than 30% with no commutation problems (transistor-based technology). Also the lower losses with DC transmission line imply longer load reach distances, as shown in Fig. 3.1 and Fig. 3.2. However, because it requires two converter stations having a total power rating of twice the maximum required power transmission, DC technology is not that inexpensive. The cost savings must, therefore, be evaluated against the cost of the two converter stations.

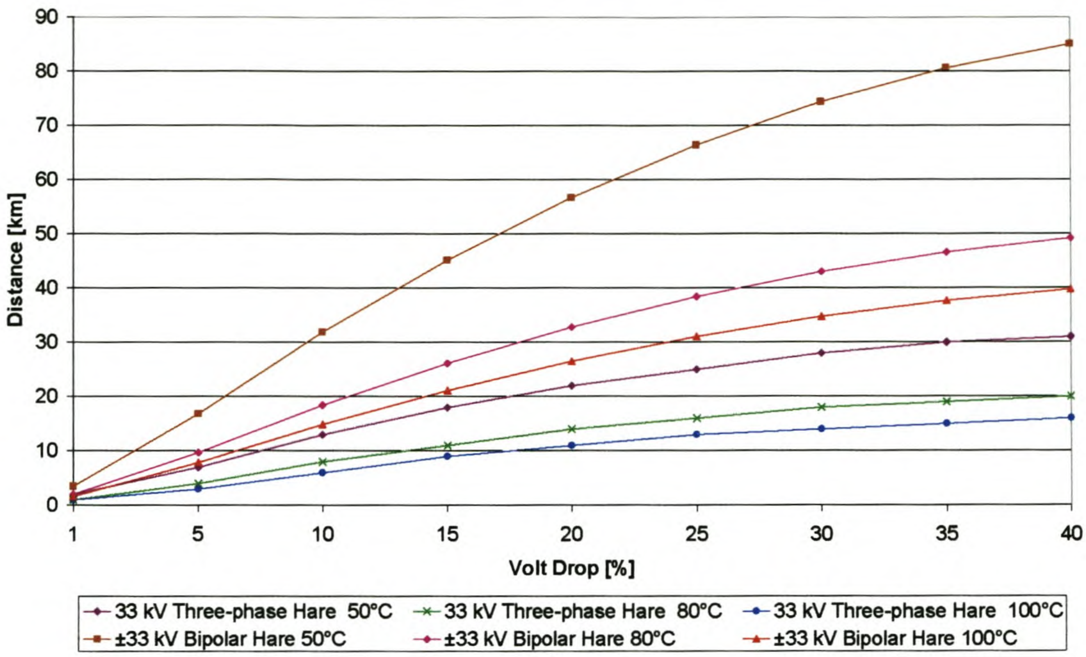


Fig. 3.3: Probabilistic load reach of 33 kV three-phase and bipole Hare at different template temperatures

Shown in Fig. 3.3 is the combined impact of increased current ratings (at template temperatures of 50, 80 and 100°C) and extended voltage-regulation margins in determining the thermal load reach of 33 kV three-phase and bipolar Hare. By increasing the template temperature the ratings of the different lines increase, but this in turn implies shorter load reach distances.

3.4 TRANSFER CAPABILITIES OF AC AND DC LINE TECHNOLOGIES AT DISTRIBUTION VOLTAGE LEVELS

An analysis has been made to illustrate the effectiveness of different MV line technologies from a load capacity [MVA] and transfer capability [MVA.km] point of view [C16].

The transfer capability of a line technology is basically the amount of power that can be transferred over one kilometer of the line, at a specific voltage regulation limit and power factor. Rewriting equation (2-127) in terms of the receiving-end power S_R and substituting line length (l) with one kilometer, the transfer capabilities of three-phase AC line technologies at distribution level can be determined as:

$$0 = c_6 S_R^2 + c_7 S_R + c_8 \tag{3-9}$$

where

$$c_6 = \left(r^2 + x^2\right) = |z|^2 \tag{3-10}$$

$$c_7 = \left(2r\cos\theta + \left(2x - b|z|^2\right)\sin\theta\right)|V_R|^2 \tag{3-11}$$

$$c_8 = \left(1 - bx + \frac{b^2\left(r^2 + x^2\right)}{4}\right)|V_R|^4 - |V_S|^2|V_R|^2 \tag{3-12}$$

Equation (3-9) can be solved with the general expression for a quadratic equation:

$$MVA \cdot km = \frac{-c_7 \pm \sqrt{c_7^2 - 4c_6c_8}}{2c_6} \tag{3-13}$$

which will yield two possible solutions. The negative solution is ignored, as the transfer capability of a line cannot be a negative number. Again Matlab was used to calculate the values for the constants $c_6 - c_8$ and the transfer capability *MVA.km*. Addendum B6 contains the listing of *transfer.m*, the file written to solve (3-13).

The transfer capability of a bipolar DC line can easily be calculated from equation (2-173), where the line length is taken as one kilometer:

$$MW \cdot km = \frac{V_R(V_S - V_R)}{r_{dc}} \tag{3-14}$$

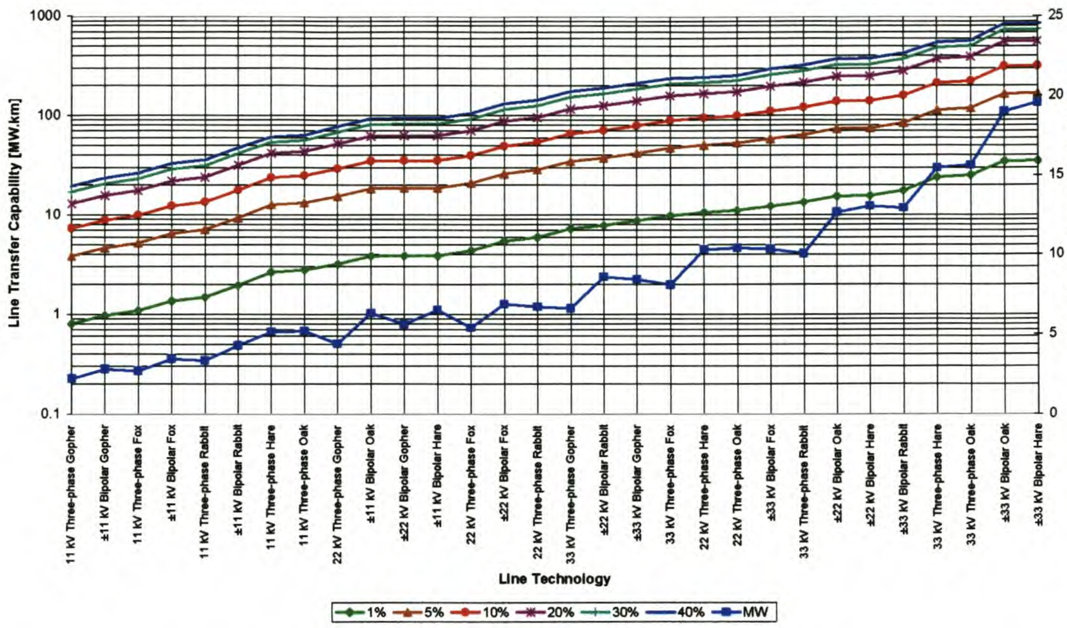


Fig. 3.4: Line transfer capabilities for different Nampower AC and DC line technologies, distribution voltages and conductor sizes

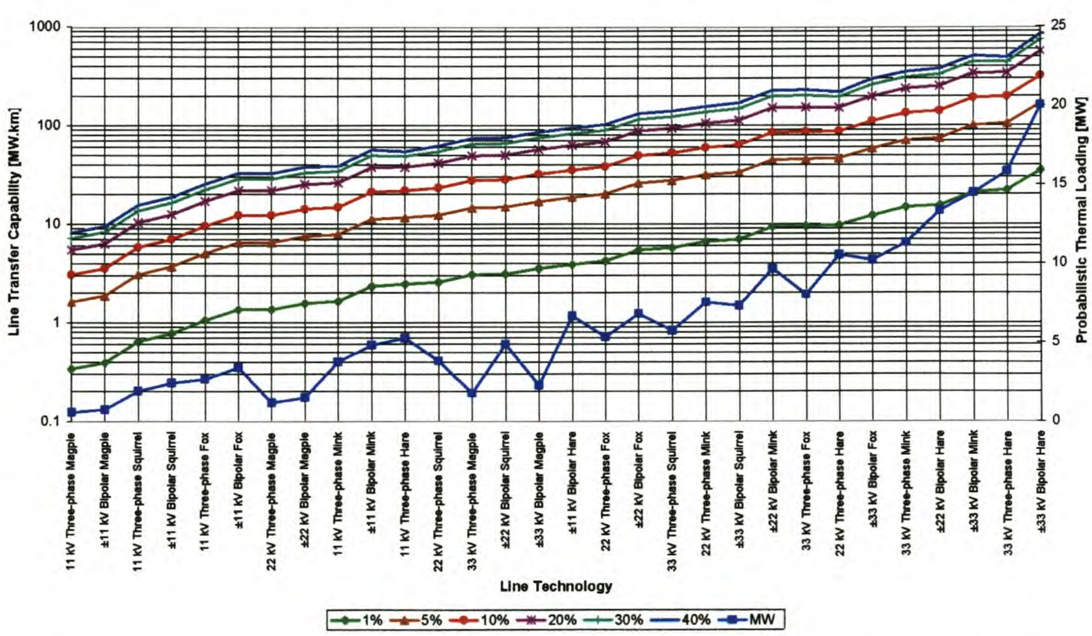


Fig. 3.5: Line transfer capabilities for different Eskom AC and DC line technologies, distribution voltages and conductor sizes

Fig. 3.4 and Fig. 3.5 summarize the line transfer capabilities of different three-phase and bipolar networks at distribution voltage levels for Nampower and Eskom conductor sizes, respectively. A power factor of 0.95 was used to calculate the real power transfer capabilities of the AC line technologies.

These graphs show that higher voltage lines can transfer more power, since power is proportional to the square of the voltage. They normally cost more due to expensive insulation, but this is offset by the much increased transfer capability [C6]. Also shown in Fig. 3.4 and Fig. 3.5 is the much higher transfer capability of bipolar DC lines for the same conductor size. Even if the voltage drop limitations of AC distribution networks is increased beyond the acceptable norm of 7.5% and 10% up to 40% [B15], DC lines will still have superior transfer capabilities over AC lines, due to lower line losses.

CHAPTER 4 THE KATIMA MULILO CASE STUDY

4.1 INTRODUCTION

Chapter 4 represents the practical application of the theory developed in Chapter 2 and Chapter 3. A case study is presented which involves the construction of a new line to transmit a small amount of power over a long distance. Two transmission technologies are evaluated, namely 132 kV three-phase AC and ±50 kV bipolar DC. It is shown that the AC network will require compensation due to overvoltages at a distance from the source and the effects of shunt, series and series-shunt compensation are considered for these networks. The DC network is evaluated as an in-line compensated AC network.

A description of the Katima Mulilo case study is presented in section 4.2. Different ACSR conductors are evaluated by TESAT in 4.2.1 to determine the optimum conductor for a 500 km, 132 kV three-phase AC line. The methods used to determine the network parameters for the generic network of PSAT are also presented in 4.2.2. A voltage regulation problem for the AC network is identified in 4.2.3 and in sections 4.3 - 4.5, shunt, series and series-shunt devices are analysed as voltage regulators. In section 4.6.1 different ACSR conductors are evaluated by TESAT to determine the optimum conductor for a 500 km, ±50 kV bipolar DC line. PSAT is then used to analyse the voltage regulation performance of the DC transmission network as an in-line technology in 4.6.2.

The chapter is concluded with a summary of the different options available for constructing the new network in order to determine the optimum solution for this case study.

4.2 THE KATIMA MULILO CASE STUDY

The Caprivi region, shown in Fig. 4.1, is a thin projection of north-eastern Namibia surrounded by the countries of Angola, Zambia and Botswana. Its environmental character derives from the three perennial rivers that cross and border the area and the flat landscape that separates them.

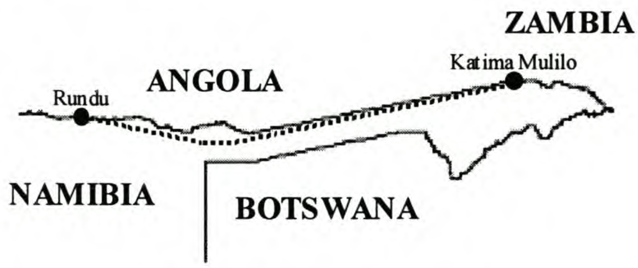


Fig. 4.1: The Caprivi region

The current total electrical load of the Caprivi Region amounts to approximately 0.52 MVA at a power factor of 0.96. This load is supplied from Katima Mulilo distribution station. The Caprivi network is isolated from the rest of the Namibian electricity network (about 500 km from the closest point of supply at Rundu). The Katima Mulilo network is currently connected to the Zambian national electricity supply authority (ZESCO). The Zambian network is heavily loaded and operates at the limits of its capacity. This results in frequent power outages. The total load of the Caprivi region is anticipated to reach 8.5 MW after 20 years, which corresponds to a load growth of 15.22% per annum [C17].

4.2.1 AC line technology optimisation

The AC interface of TESAT is used to evaluate three different Nampower conductor choices in order to select the optimum conductor for a 500 km, 132 kV three-phase AC line (shown in Fig. 4.1) between Rundu substation and Katima Mulilo distribution station. The following three line technologies were evaluated:

- Wolf conductor with a TRANS 980 tower
- Pelican conductor with a Concrete tower
- Twin Hare conductor with a Kamerad tower.

The line structures used by Nampower for the above-mentioned conductors are shown in Fig. 4.2. The data for the different conductors were sourced from conductor property tables and are included in Addendum A1.

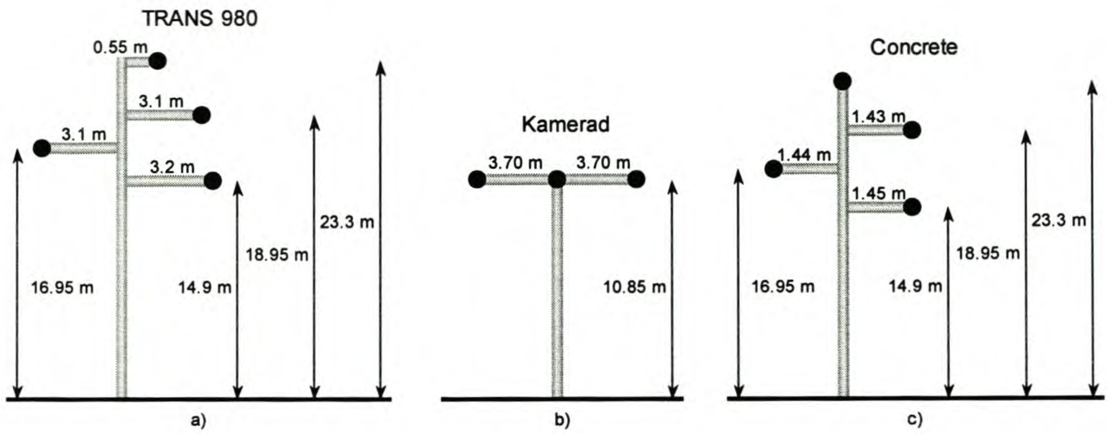


Fig. 4.2: 132 kV Nampower line structures [C18]

The series impedance and shunt admittance of the different conductors and line structures were calculated, using the OLP interface of PSAT. The RXB values for the different line technologies are given in Table 4-1, as well as the deterministic ampacity values and the cost of these lines. The AC resistance values at 40°C were calculated from the DC resistance values at 20°C.

Table 4-1: 132 kV Nampower line technology data

Conductor	Tower	$r_{ac\,(40°C)}$ [Ω/km]	X [Ω/km]	B [μS/km]	Ampacity [A]	Line Cost [R/km]
Wolf	TRANS 980	0.2091	0.4157	2.72	372	127 000
Pelican	Concrete	0.1291	0.3665	3.13	486	151 000
Twin Hare	Kamerad	0.1601	0.3041	3.73	571	179 000

The data in Table 4-1 are entered in the AC interface of TESAT. A base value of 100 MVA is used for S_{base} and 132 kV for V_{base}. Eskom supplied the generation cost for the next 20 years, where the year 1999 was taken as the base year. The current interest rate was taken as 13.8% and the ALF as 0.4. Nampower supplied the line costs, which are treated as confidential and are used only for reference purposes. The costs of the line technologies per kilometre were calculated using a cost estimation tool provided by Eskom. These costs included material costs, construction costs, survey costs, field services, engineering fees and general overhead costs.

Voltage regulation analysis

The voltage plots indicating the voltage regulation of the different line technologies are shown in Fig. 4.3. The growth in point load is plotted on the X-axis and the physical position along the line on the Y-axis. Therefore, the Z-axis represents the voltage magnitude. These graphs can be used for additional insight into the voltage regulation of different line technologies for different point load sizes and distances. From Fig. 4.3 it is clear that the three-phase line technologies will suffer from voltage regulation problems.

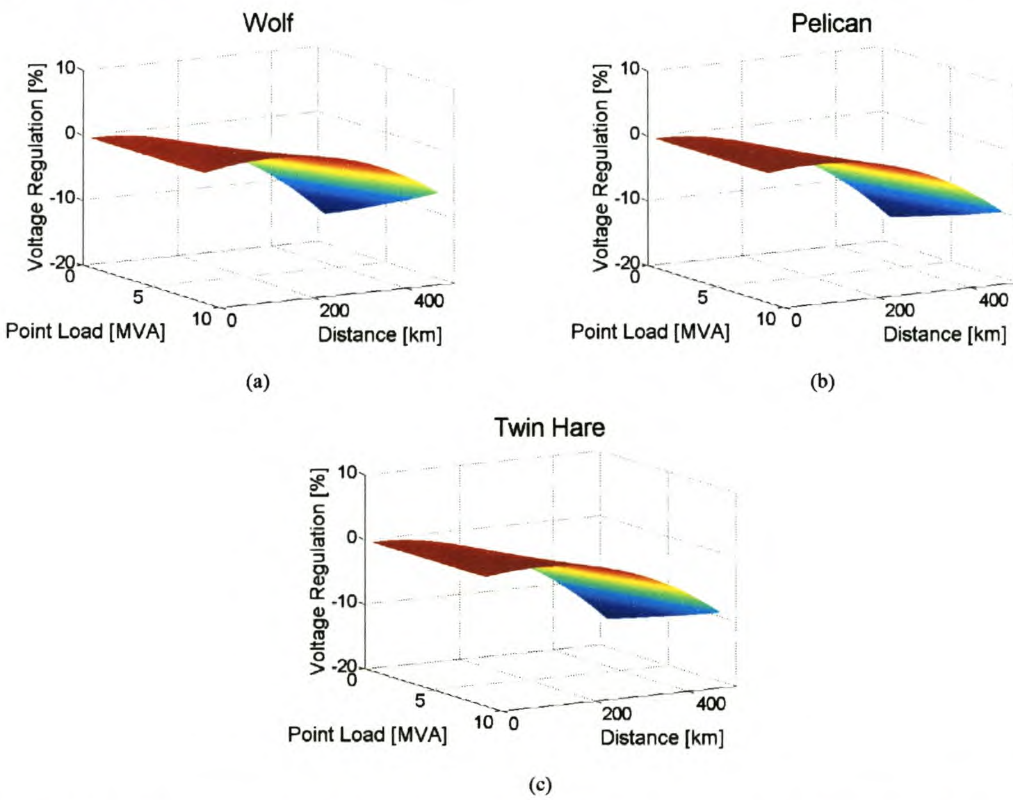


Fig. 4.3: Voltage regulation of the AC line technologies as a function of distance and load growth

A cut parallel to the XZ-plane at a distance of 500 km (Fig. 4.4) reveals the true extent of the voltage regulation problem. Due to the long distance and the small point load, there is a rise in the receiving-end voltage, due to the flow of capacitive (line-charging) current through line inductance (Ferranti effect). The worst-case scenario is a sudden opening of the line at the receiving end, which will cause the sending-end voltage to rise due to the capacitive current of the line flowing through the source impedance. This worst-case over-voltage is approximately 16% (Fig. 4.4).

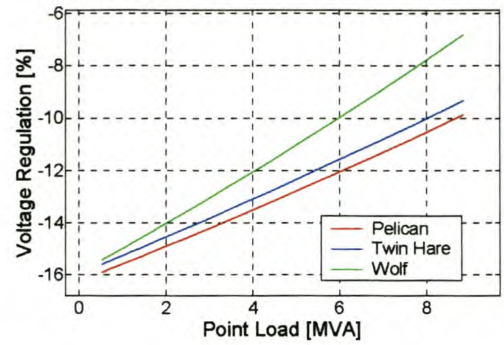


Fig. 4.4: Voltage regulation of the AC line technologies as a function of load growth

Table 4-2 provides a summary of the voltage regulation performance for the different AC line configurations in Fig. 4.4. The values in Table 4-2 are well above the NRS 048 standard of 1.05 p.u. for ≥ 500 V networks [B16]. Appropriate forms of line compensation must therefore be provided to keep the overvoltages on the lines to acceptable limits.

Table 4-2: Summary of the voltage regulation performance

132 kV	% Voltage Regulation	
Three-phase AC	Minimum	Maximum
Pelican	-9.8	-15.9
Twin Hare	-9.3	-15.6
Wolf	-6.8	-15.4

A cut parallel to the YZ-plane gives the voltage regulation at different positions along the line for a specific load. The voltage regulation for an initial load of 0.52 MVA and a final load of 8.85 MVA is shown in Fig. 4.5. Fig. 4.5 shows that, as the load grows from its initial value to its final value, there will be an improvement in the voltage regulation along the line. Depending on the load current, the lines will either absorb or supply reactive power. At loads below the SIL the lines will produce net reactive power. Increasing the load increases the line current that flows from the source to the load, decreasing the net effect of the line charging current and improving the voltage regulation along the line.

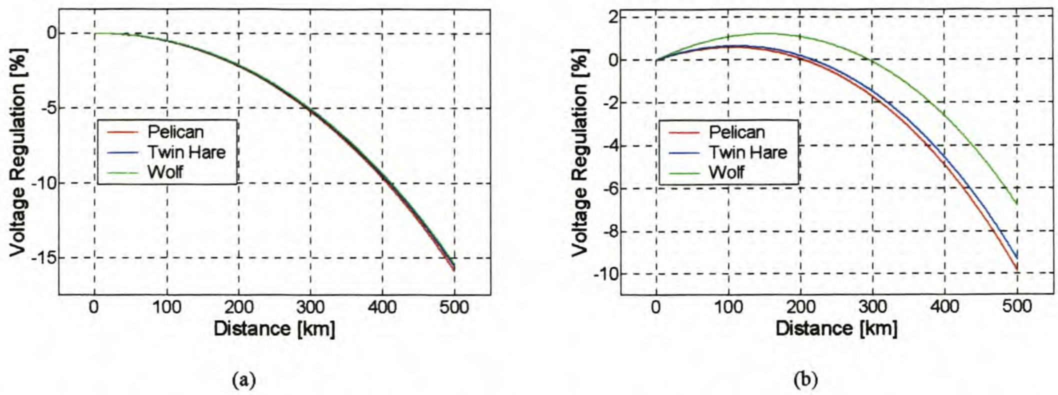


Fig. 4.5: Voltage regulation of the AC line technologies as a function of position for an (a) initial load of 0.52 MVA and (b) a final load of 8.85 MVA

The voltage regulation of the different line technologies can also be explained with the aid of a contour plot, which is a cut parallel to the XY-plane. Fig. 4.6 shows the voltage regulation contours of the different line technologies for a value of zero regulation. The growth in point load is plotted on the X-axis and the position along the line on the Y-axis. The value for voltage regulation will be negative (rise in receiving-end voltage) for all the combinations of point load and position above the voltage regulation contours of the different line technologies. For all the values below the contours the voltage regulation will be positive.

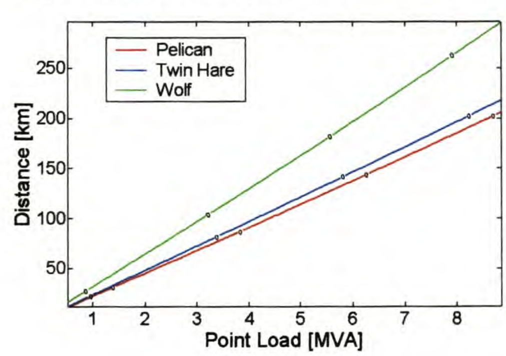


Fig. 4.6: Contour plot for voltage regulation of the AC line technologies

As for voltage regulation, TESAT provides similar three- and two-dimensional graphs for power losses, transmission costs and ATIs. The graphs were analysed in detail for voltage regulation, and from this point forward only the graphs portraying voltage regulation, losses, transmission costs and ATIs against growth in point load will be studied.

Transmission losses

The power losses in kW for the different three-phase line technologies are shown in Fig. 4.7 and do not include compensation losses. The lines will experience maximum power losses for initial load conditions. The maximum losses for the different lines are more than 100% of the transmitted power and are due to large line charging currents flowing through the line inductance for small loads over long distances. As the load increases, the line charging-currents and the power losses decreases to a minimum value that differs for the three cases shown in Fig. 4.7.

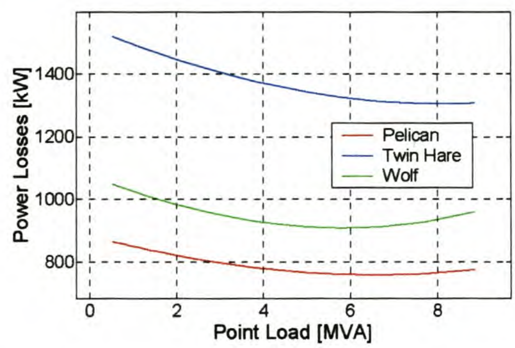


Fig. 4.7: Power losses of the AC line technologies as a function of load growth

Table 4-3 provides a summary of the power losses for the different AC line configurations in Fig. 4.7. The maximum losses are between 150% and 260% of the transmitted power, whereas the minimum losses are between 11% and 17% of the transmitted power.

Table 4-3: Summary of power losses

132 kV Three-phase AC	Minimum		Maximum	
	$P_{phase\ loss}$ [kW]	$P_{line\ loss}$ [kW]	$P_{phase\ loss}$ [kW]	$P_{line\ loss}$ [kW]
Pelican	253	760	289	867
Twin Hare	436	1 307	507	1 522
Wolf	303	910	350	1 050

Transmission costs

Fig. 4.8 shows the transmission costs or LCC of the different three-phase line technologies in terms of the present value of money. The cost of line compensation required for the AC line technologies is not included in the LCC of Fig. 4.8. The three-phase Wolf proves to be the most economic solution for the lifetime of the AC network (20 years), while Twin Hare will be the most expensive solution.

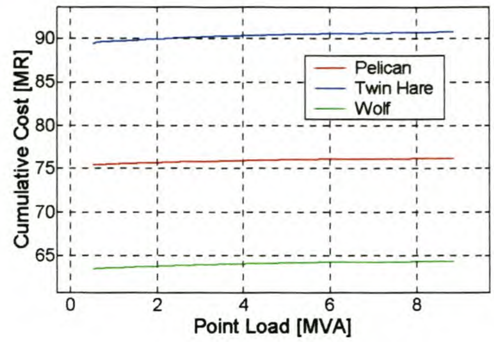


Fig. 4.8: LCC for the AC line technologies

Table 4-4 provides a summary of the minimum and maximum present values of the LCC over the lifetime of the uncompensated AC network. The present value of the cost of losses is also shown in Table 4-4, where Pelican will have the lowest cost of losses and Twin Hare the highest cost of losses.

Table 4-4: Summary of LCC and cost of losses

132 kV Three-phase AC	PV_{LCC} [MR]		PV_{TCL} [MR]
	Minimum	Maximum	
Pelican	75.50	76.27	0.77
Twin Hare	89.50	90.85	1.35
Wolf	63.50	64.43	0.93

Appropriate technology index (ATI)

The ATI technique is used to identify the optimum conductor for the three-phase line. As was discussed previously in section 2.7.1., the KPIs to consider for HV networks are LCC, capacity, stability and voltage regulation. The capacity KPI is ignored, because the load current requirement is much smaller than the current rating of the conductors used to supply the load. Owing to small point load, the power angle across the line is small (about 9°), and as a result the stability KPI is also ignored. All the line technologies will require voltage compensation during the lifetime of the network and, therefore, the weighting factors for the LCC and voltage regulation are chosen to be 50% each.

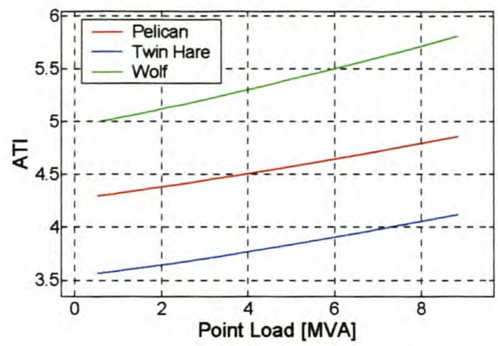


Fig. 4.9: ATI of the three-phase line technologies

The ATIs for the three-phase line technologies are shown in Fig. 4.9 and the results are summarised in Table 4-5. Fig. 4.9 clearly indicates Wolf conductor to be the optimum choice for the lifetime of the line. On the contrary, Twin Hare conductor would be the least optimum choice to string the line with.

Table 4-5: ATI summary of the three-phase line technologies

132 kV Three-phase AC	ATI Minimum	ATI Maximum
Pelican	4.3	4.9
Twin Hare	3.6	4.1
Wolf	5.0	5.8

4.2.2 Obtaining network parameters

The Caprivi network can connect to the Namibian electricity network at Rundu substation at 132 kV. Nampower supplied the source impedance at the 132 kV and 66 kV busbars at Rundu substation and at the 66 kV busbar at Katima Mulilo distribution station. These values are shown in Table 4-6 below. The high source impedance at the 132 kV Rundu busbar is of particular concern. This high value is due to the long lines that characterise the Namibian electricity network. A switched 7 MVar shunt reactor, with a corresponding reactance of j2489.89 Ω, is connected at the 132 kV Rundu busbar to compensate for a voltage rise at Rundu substation during light loading conditions. The connection of another 500 km line with a relatively small point load at this weak point will just worsen the voltage regulation problems, unless some form of compensation is provided to reduce the source impedance.

Table 4-6: Source impedances

Bus	Voltage [kV]	Impedance [Ω]
Rundu	132	85.23 + j214.34
Rundu	66	25.70 + j77.52
Katima Mulilo	66	58.68 + j143.27

The source impedance is compensated for using a series capacitor and two identical shunt reactors connected on either side of the capacitor as shown in (a) of Fig. 4.10. The purpose of the series capacitor is to reduce the source inductance, while the purpose of the shunt reactors is to control the line voltage on either side of the series capacitor [B17]. The series compensating capacitor was chosen to compensate for 50% of the source reactance, so that X_C = j107.17 Ω (450 kVar). X_{R1} is the 7 MVar switched busbar reactor at Rundu substation. To save costs this reactor is reconfigured as a line reactor on the one side of the series capacitor and a second identical line reactor, X_{R2}, is added on the other side. All values in Fig. 4.10 are per unit, with base values S_{base} = 100 MVA and V_{base} = 132 kV.

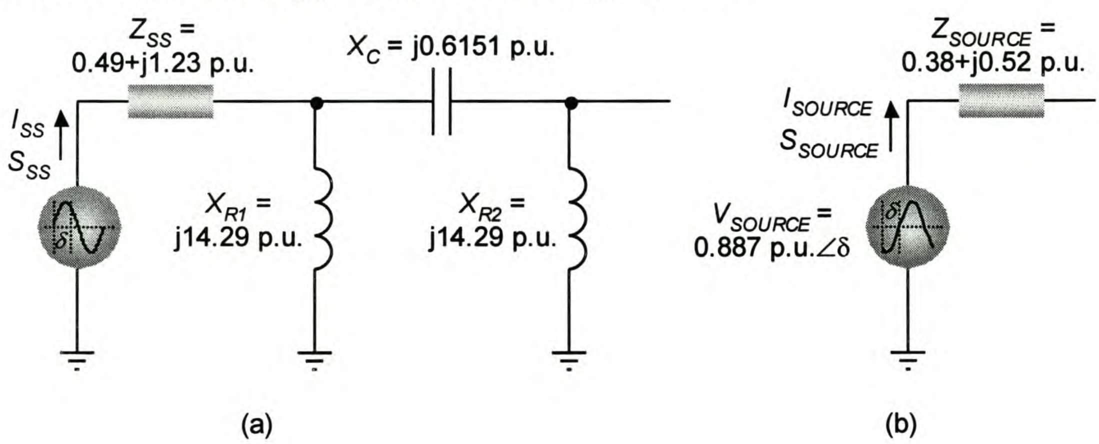


Fig. 4.10: Compensated source

The model in (a) of Fig. 4.10 can be simplified by Thévenin-Norton source transformations until the network corresponds to (b), where V_{SOURCE} and Z_{SOURCE} are calculated as follows:

$$V_{SOURCE} = \frac{X_{R1}X_{R2}V_{SS}}{\left(Z_{SS} + jX_{R1}\right)\left(\dfrac{jZ_{SS}X_{R1}}{Z_{SS} + jX_{R1}} - jX_C + jX_{R2}\right)} \tag{4-1}$$

$$Z_{SOURCE} = \frac{j\left(\dfrac{jZ_{SS}X_{R1}}{Z_{SS}+jX_{R1}} - jX_C\right)\cdot X_{R2}}{\dfrac{jZ_{SS}X_{R1}}{Z_{SS}+jX_{R1}} - jX_C + jX_{R2}} \tag{4-2}$$

From section 4.2.1 it was determined that Wolf conductor with a TRANS 980 structure would be the optimum conductor-tower combination for the AC network. The series impedance and shunt admittance was calculated previously by the OLP interface of PSAT for the specified conductor and line structure. These values are then used to calculate the ABCD parameters, which is an exact representation of the line. It would, however, be more convenient to have a circuit representation of the line. The equivalent π circuit model is calculated in the OLP interface to have identical ABCD parameters to what was calculated previously for the specific conductor and line structure.

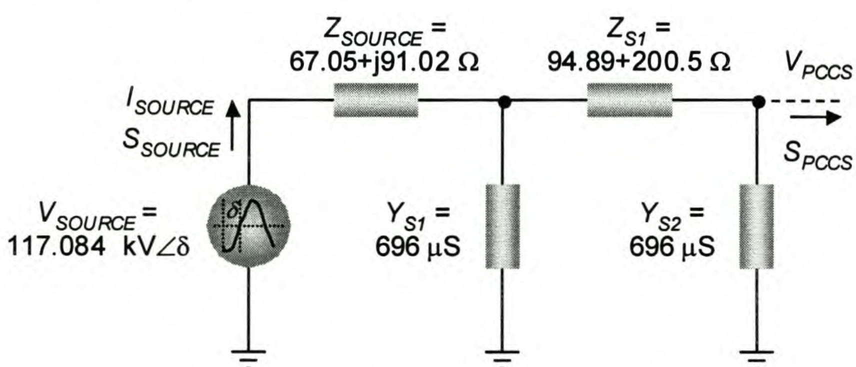


Fig. 4.11: Compensated source impedance and equivalent π circuit model

The compensated source and equivalent π circuit model are shown in Fig. 4.11. Z_{S1} and Y_{S1} are the total series impedance and shunt admittance of the line, respectively. This model is implemented in OLP and can be simplified by one or two Thévenin-Norton source transformations. The results, V_S and Z_S, are converted to per unit values and exported to the sending-end of PSAT. Equations (4-3) and (4-4) are used to calculate V_S and Z_S:

$$V_S = \frac{V_{SOURCE}Z_{S11}Z_{S12}}{\left(Z_{S11}Z_{SOURCE} + Z_{S1}Z_{SOURCE} + Z_{S1}Z_{S11} + Z_{S12}Z_{SOURCE} + Z_{S12}Z_{S11}\right)} \tag{4-3}$$

$$\begin{aligned} Z_S &= \frac{Z_{S12}\left(Z_{SOURCE}Z_{S11} + Z_{SOURCE}Z_{S1} + Z_{S12}Z_{S11}\right)}{\left(Z_{S11}Z_{SOURCE} + Z_{S1}Z_{SOURCE} + Z_{S1}Z_{S11} + Z_{S12}Z_{SOURCE} + Z_{S12}Z_{S11}\right)} \\ &= R_S + jX_S \end{aligned} \tag{4-4}$$

where $Z_{S11} = Y_{S1}^{-1}$ and $Z_{S12} = Y_{S2}^{-1}$.

4.2.3 Uncompensated network

The simplified model of the complete network, including the compensated source, the 500 km line and the initial load at Katima Mulilo, is shown in Fig. 4.12. This is a radial network and has only one voltage source, namely Rundu substation. A voltage-regulation problem is anticipated for initial load conditions. Due to the long distance and small point load, there is a rise in the receiving-end voltage, due to the flow of capacitive (line-charging) current through line inductance (Ferranti effect).

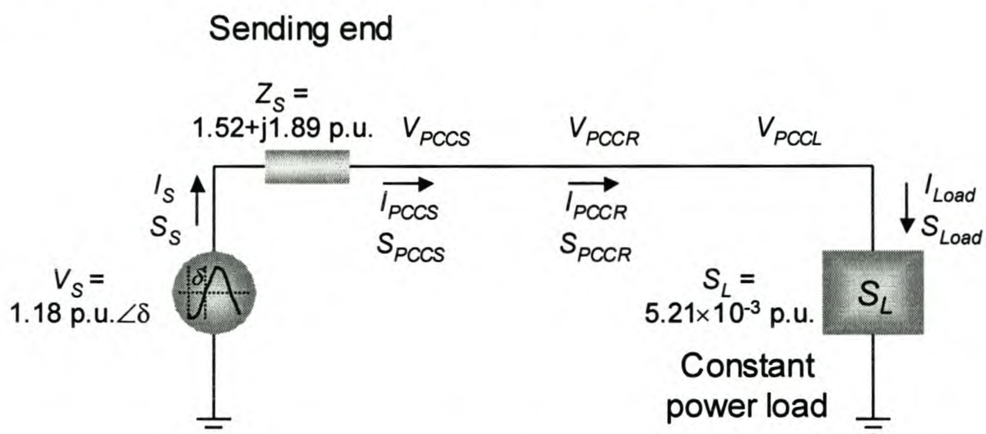
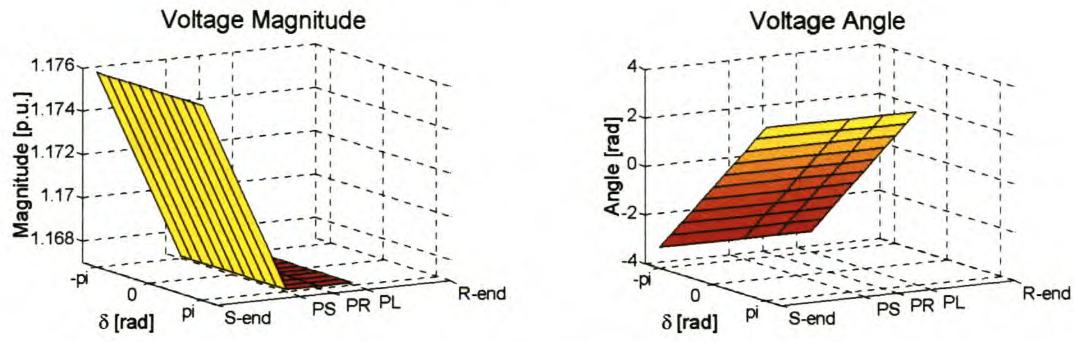


Fig. 4.12: Equivalent circuit diagram for the Katima Mulilo network

The circuit in Fig. 4.12 was analysed in PSAT. The phase angle of V_S (δ, as indicated in Fig. 4.12) was given a fixed value when the network parameters were determined. All other parameters were kept constant while δ was varied from -π to π to simulate various voltage angles across the network. A higher δ generally results in a higher power transfer across a network, but in practice uncompensated networks are seldom operated at power angles larger than 0.7 rad. The possibility exists that a compensated network can be operated at larger voltage angles because the compensator can increase the first swing stability margin [B3].

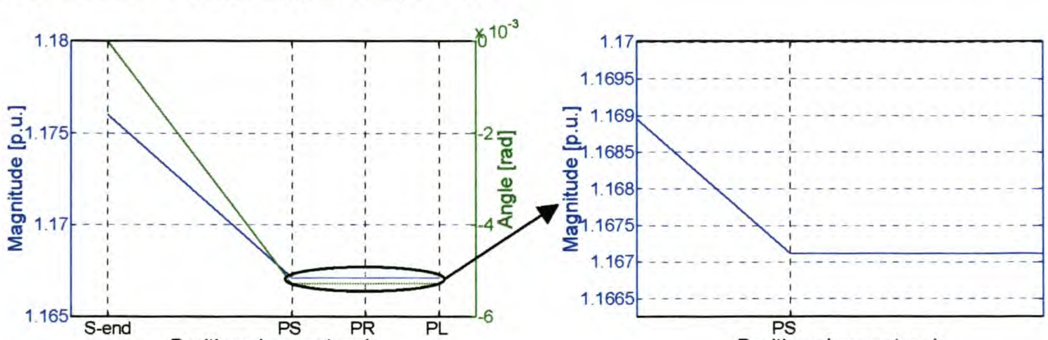
The voltage plots indicating the voltage regulation for this network are shown in Fig. 4.13. The figure shows the voltage regulation across the network of Fig. 4.12. The physical location on the network is plotted on the Y-axis and δ is plotted on the X-axis. Therefore, the Z-axis represents the respective voltage magnitude and angle at a specific location in the network, with a certain phase angle across the network. The graphs are plotted for $-\pi \leq \delta \leq \pi$.



(a) (b)

Fig. 4.13: Voltage regulation as a function of network position and voltage angle for the uncompensated Katima Mulilo network

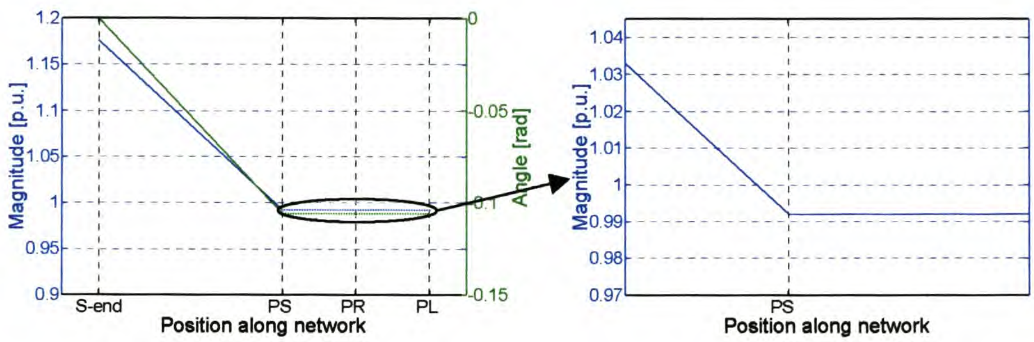
The voltage regulation problem at the point of common coupling (PCC) is clearly visible in the magnitude plot in (a). The sending end, receiving end and the load are situated at the PCC. In all plots the PCC is labelled as follows: PS for PCCS (sending-end side of PCC), PR for PCCR (receiving-end side of PCC) and PL for PCCL (load side of PCC). Due to the constant load at Katima Mulilo and only one voltage source in the network, the voltage magnitude at a specific location in the network remains constant for all δ. For no receiving-end source or impedance, the voltage graphs stop at the PCC. The voltage drop across Z_S is visible between the sending end and PCCS in both the voltage magnitude (a) and angle (b) graphs. For a radial network the relative voltage angle across the network stays constant for all δ. The difference in the voltage angles between the sending end and the PCC is constant for all δ and is due to the voltage drop across Z_S. The third dimension therefore contains no information. A cut parallel to the YZ-plane at $\delta = 0$ reveals the true extend of the voltage regulation problem.



(a) (b)

Fig. 4.14: Voltage regulation as a function of network position for the uncompensated Katima Mulilo network and an initial load of 5.21 x 10^{-3} p.u.

In Fig. 4.14 $|V_{PCCR}|$ equals 1.167 p.u., which is 0.117 p.u. above NRS 048 standard of 1.05 p.u. for >500 V networks. $|V_{PCCS}|$ also equals 1.167 p.u. (V_{PCCS} and V_{PCCR} are physically at the same point for the uncompensated network), which is not acceptable for loads along the line. In this case no loads are connected along the line, but the line equipment is stressed, which in turn reduces its lifespan. This increases the probability of flashovers and failures. A higher load demand will improve the voltage regulation at the PCC.



(a) (b)

Fig. 4.15: Voltage regulation as a function of network position for the uncompensated Katima Mulilo network and a final load of 88.54 x 10^{-3} p.u.

In Fig. 4.15 $|V_{PCCR}|$ and $|V_{PCCS}|$ equals 0.992 p.u which is within NRS 048 limits. The voltage regulation of the anticipated final load for the Katima Mulilo network, as well as all the points along the line, will be within the required limits. No compensation, therefore, will be required for higher load demands. The negative voltage angle at the PCCR is due to the voltage drop across Z_S while the sending-end voltage angle, δ, is zero.

Transmission cost of the uncompensated network

The Caprivi network is supplied from Katima Mulilo distribution station at 66 kV. An 10 MVA 132/66 kV transformer must, therefore, be installed at Katima Mulilo distribution station and the cost included in the LCC of the uncompensated network. The equipment needed to reduce the source impedance at Rundu substation also increases the capital cost of the uncompensated network. These capital costs are summarised in Table 4-7.

Table 4-7: Additional capital costs incurred for the uncompensated Katima Mulilo network

Description	Total [R]
10 MVA 132/66 kV transformer	1 112 000
450 kVar capacitor bank	4 800
Switchgear, civil work and steelwork	1 500 000
Primary over-voltage protection	1 200 000
Control and protection	210 000
7 MVar shunt reactor bank	700 000

Nampower supplied the costs of the transformer and the shunt reactor bank. The costs of the series capacitor and the equipment to protect the capacitor against fault conditions, as well as other related costs, were supplied by ABB. The present value of LCC (PV_{LCC}) for the lifetime of the uncompensated Katima Mulilo network is shown in Fig. 4.16

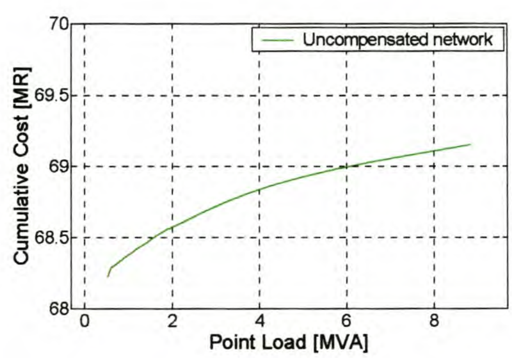


Fig. 4.16: LCC of the uncompensated Katima Mulilo network

A summary of the minimum and maximum present values of LCC over the lifetime of the uncompensated AC network is shown in Table 4-8. PV_{LCC} included the capital investment cost of the 500 km, 132 kV three-phase Wolf line, the transformer at Katima Mulilo distribution station and the source compensation equipment at Rundu substation. Only the cost of losses in the line (PV_{TCL}) was included in PV_{LCC}, whereas the cost of losses in the transformer and the equipment required for source compensation were not included.

Table 4-8: Summary of the LCC of the uncompensated Katima Mulilo network

Network Configuration	PV_{LCC} [MR]		PV_{TCL} [MR]
	Minimum	Maximum	
Uncompensated	68.23	69.16	0.93

The voltage regulation problems and the high cost of the equipment needed to protect the series capacitor during fault conditions raises suspicions about the effectiveness of an AC interconnection in terms of cost and technical performance. In the sections to follow shunt, series and series-shunt compensator topologies will be investigated and analysed as solutions to the voltage regulation problem of the AC network for small loads. DC transmission, as an alternative to an AC interconnection, will also be investigated in order to determine the optimum solution for this network.

4.3 SHUNT COMPENSATION

Katima Mulilo has a 3.75 MVA diesel-fired power station, which consists of six generators with a capacity of 0.5 MW and 0.375 MVar each. These generators only turn in during outages. Using the generators as synchronous condensers may be a solution to the voltage regulation problem for light loading conditions. The generators will be under-excited in order to absorb reactive power. Used in this manner, the generators inject no active power into the network and therefore do not expend any fuel (no energy storage required). With the generators connected directly to the AC network, reaction time is in the order of cycles and continuously variable compensation is achieved.

An alternative solution is to install shunt reactors, which will absorb purely reactive power. The shunt reactors are used to provide inductive reactance to control system voltage and to compensate for the effects of the high charging current of long lines. For light load conditions this charging current can produce more leading reactive power than the system can absorb without risk of instability or excessively high voltages at the line terminals [A13]. Apart from being switched, these devices are incapable of continuous variable compensation. An intelligent controller, which monitors the voltage waveform, can be used to switch the reactors when the network voltage exceeds the NRS 048 specifications. Such a controller can reduce switching cycles to milliseconds. Switching transients introduced by the reactor switching are also a major drawback of this method.

Fig. 4.17 is the circuit diagram of the shunt-compensated network. $|V_{PCCR}|$ is specified as 1.0 p.u. to adhere to NRS 048 specification. Note that the voltage angle at the sending end is now taken as the reference angle (0 rad), while the voltage angle at the PCC is varied from $-\pi \leq \delta \leq \pi$. The network parameters are known from Fig. 4.12.

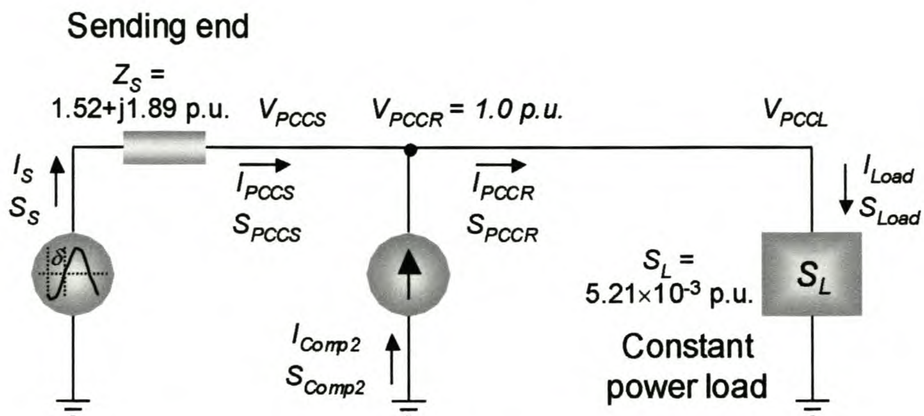


Fig. 4.17: Katima Mulilo network with a shunt compensator for light loads

The voltage diagram generated with PSAT at the PCCR and PCCS, which are the same points for shunt compensation, is shown in Fig. 4.18. $|V_{PCCR}|$ is lowered to 1.0 p.u. for all δ, which is how the shunt compensator is expected to work. Fig. 4.18 shows that the voltage regulation for light loads, as well as all the points along the line, will be within the required limits. As for the uncompensated network, the difference in voltage angles is constant for all δ and is due to the voltage drop across Z_S.

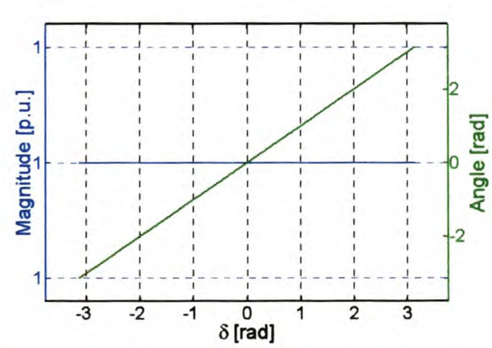
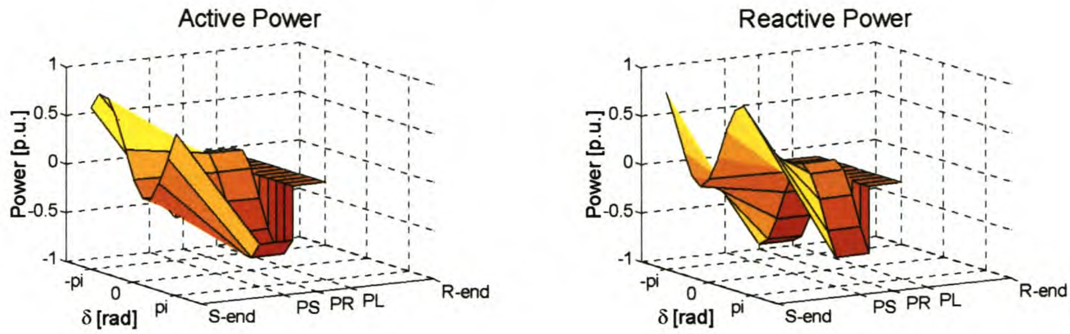


Fig. 4.18: Voltage plots at the PCC

Although phase-shifted by the losses in the network, the reactive power demand at the sending end is a typical co-sinusoidal function of δ, while the active power demand is a sinusoidal function of δ, as shown in Fig. 4.19. The chosen direction of power flow is shown in Fig. 4.17.



(a) (b)

Fig. 4.19: Active and reactive power dissipated across the shunt-compensated Katima Mulilo network

The shunt compensator, by adding its active and reactive power to the network, creates the step in active and reactive power at the PCCR. The step at the PCCL in the active and reactive power graphs is the active and reactive power dissipated by the load at Katima Mulilo distribution station. PSAT shows the compensator power as a separate two-dimensional graph, shown in Fig. 4.20.

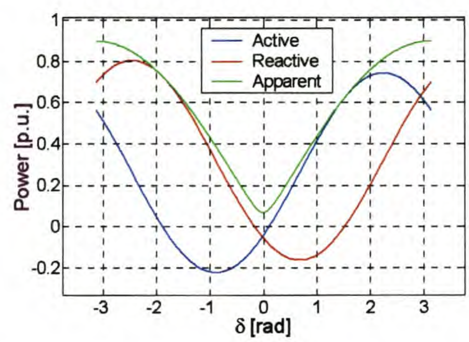


Fig. 4.20: Power plots for the shunt compensator

The compensator has a minimum rating of 0.0683 p.u. (6.83 MVA). The active and reactive power absorbed are 0.0404 p.u. (4.04 MW) and 0.0551 p.u. (5.51 MVar), respectively. This minimum rating is achieved at $\delta = 0$ rad. This is well above the capability of the generators at the diesel-fired power station. Also, minimum compensation requires a power factor of -0.59, whereas the generators can only operate down to a power factor of 0.8.

A purely reactive compensator can be implemented when the active power is zero. This occurs at $\delta = -1.88$ rad and $\delta = 0.1$ rad. The latter is chosen as this delivers the lowest compensator rating of 0.086 p.u. A switched reactor bank capable of absorbing 8.6 MVar at the PCCR is therefore required for this control strategy.

Alternatively, the generators at Katima Mulilo power station can be modified to operate as synchronous condensers. The cost involved will be that of modifying the generators to operate as synchronous condensers. Switched shunt reactor banks can be installed to absorb the bulk of reactive power, whereas the generators can be used to rapidly compensate the balance of the required reactive power. The trade-off is the reduction in the capital investment of providing the equipment to absorb the reactive power.

Transmission losses of the purely reactive shunt-compensated network

Fig. 4.21 shows the effect of purely reactive shunt compensation on the power losses of the uncompensated network. For the uncompensated network high line charging currents produce more leading reactive power than the system can absorb, causing high power losses and excessively high voltages at the line terminals. The effect of purely reactive shunt compensation is to absorb the balance of the leading reactive power in the network, reducing power losses and overvoltages in the network.

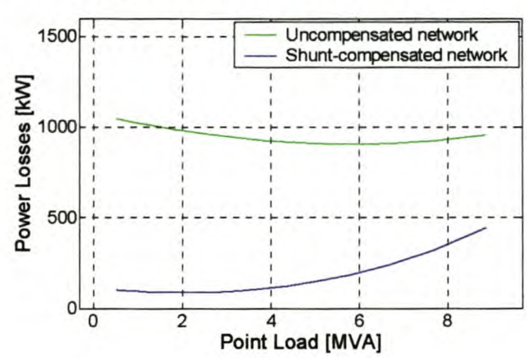


Fig. 4.21: Effect of purely reactive shunt compensation on the power losses of the uncompensated network

A shunt reactor bank of 7 MVar is installed to absorb the bulk of reactive power. The generators at Katima Mulilo power station are reconfigured to operate as synchronous condensers to absorb the rest of the reactive power (maximum of 1.6 MVar). Table 4-9 provides a summary of the total power losses for uncompensated and shunt compensated network in Fig. 4.21. The minimum losses for the shunt-compensated network are 90% less than the minimum losses for the uncompensated network, whereas the maximum losses for the shunt-compensated network are 58% less. In Fig. 4.21 the additional power losses in the shunt reactors banks and synchronous condensers were not included in the total power losses for the shunt-compensated network.

Table 4-9: Summary of total power losses for uncompensated and shunt-compensated Katima Mulilo network

Network Configuration	Minimum		Maximum	
	$P_{phase\ loss}$ [kW]	P_{Losses} [kW]	$P_{phase\ loss}$ [kW]	P_{Losses} [kW]
Uncompensated	303	910	350	1 050
Shunt compensated	30	90	148	444

Transmission cost of the purely reactive shunt-compensated network

Using shunt reactors and static condensers to compensate for overvoltages imply increased capital cost incurred. The cost of losses must therefore be used as outflows of capital in determination of the minimum present value of LCC. Fig. 4.22 shows the present value of LCC for the uncompensated and shunt-compensated networks for an ALF of 0.4. Nampower supplied the cost of the 7 MVar shunt reactor banks (Table 4-7). The cost of reconfiguring the generators as synchronous condensers is much less than the capital cost of the shunt reactor banks and was, therefore, not included.

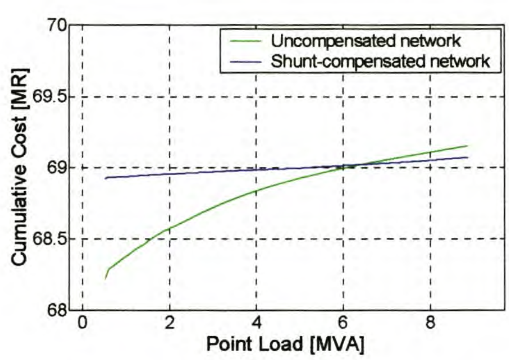


Fig. 4.22: LCC of the uncompensated and shunt-compensated Katima Mulilo network (ALF = 0.4)

From an economic point of view it can be concluded from Fig. 4.22 that only in the 18th year does the shunt-compensated network become more cost effective than the uncompensated network. Thus, only after 18 years are the savings in the cost of losses with the shunt compensated network great enough to pay for the shunt reactors.

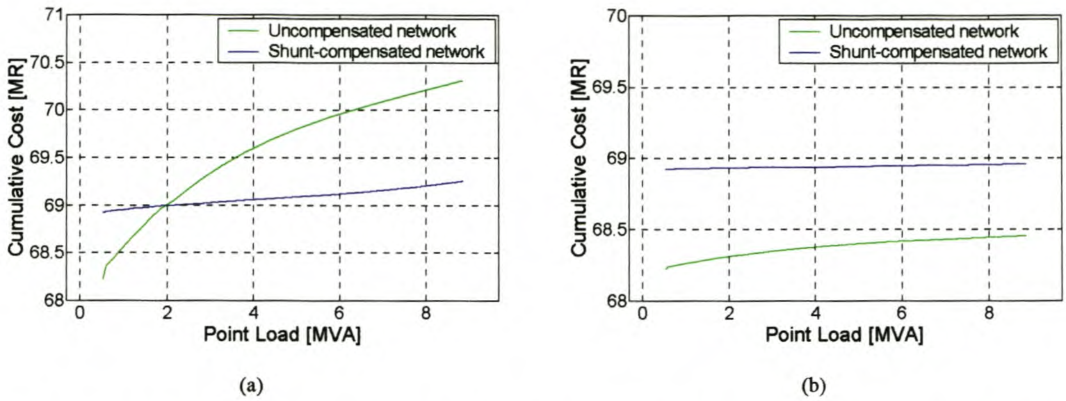


Fig. 4.23: LCC of the uncompensated and shunt-compensated Katima Mulilo network for an ALF of (a) 0.6 and (b) 0.2

The effect of increasing the ALF to 0.6 is shown in Fig. 4.23 (a). Increasing the load factor will reduce the number of years to pay back the initial capital investment of shunt compensation. In this case the payback period will be reduced from 18 years to 10 years. Similarly by reducing the ALF to 0.2 will increase the payback period. This is shown in Fig. 4.23 (b). Table 4-10 provides a summary of the minimum and maximum present values of the LCC over the lifetime of the uncompensated and shunt-compensated AC networks for different values of ALF.

Table 4-10: Summary of the LCC of the uncompensated and shunt-compensated Katima Mulilo network for different values of ALF

Network Configuration	ALF	PV_{LCC} [MR]		PV_{TCL} [MR]
		Minimum	Maximum	
Uncompensated	0.2	68.23	68.46	0.23
Uncompensated	0.4	68.23	69.16	0.93
Uncompensated	0.6	68.23	70.32	2.09
Shunt compensated	0.2	68.93	68.96	0.04
Shunt compensated	0.4	68.93	69.07	0.15
Shunt compensated	0.6	68.93	69.26	0.33

4.4 SERIES COMPENSATION

Series compensators are widely recognised as voltage regulators. A series controller could be a variable impedance, such as capacitor, reactor, etc., or a power electronics-based variable source of main frequency, subsynchronous and harmonic frequencies (or a combination) to serve the desired need [B3]. In principle all series compensators inject a voltage in series with the line. Even variable impedance multiplied by the current flow through it represents an injected series voltage in the line. The drawback of series compensation is that full load current flows through the compensator, which can be problematic under fault conditions.

The standby generators at Katima Mulilo power station cannot be used for series compensation by itself. It can, however, be used as an energy source connected through a regulator to the DC bus of a transistor-based series device. Depending on the rating of the device, this can be a very costly option. Series capacitors are a viable alternative to shunt compensation due to their relatively low cost. Depending on the controller, series capacitors are also limited by their switching times, as discussed in section 4.3. Since series capacitors compensate for the line impedance, they are self-regulating in terms of load variations. Switching is, however, often required for stability reasons. Increasing in cost and complexity, a Thyristor-Controlled Series Capacitor (TCSC) or a transistor-based device, namely a Static Synchronous Series Compensator (SSSC), can also be used as a voltage regulator [B3]. The TCSC can supply only reactive power with a reaction time of half a cycle, while the SSSC, if equipped with energy storage, can supply both active and reactive power in less than a millisecond [C1].

Fig. 4.24 is the circuit diagram of the series-compensated network. $|V_{PCCR}|$ is specified as 1.0 p.u. to adhere to NRS 048 specification. The network parameters are known from Fig. 4.12.

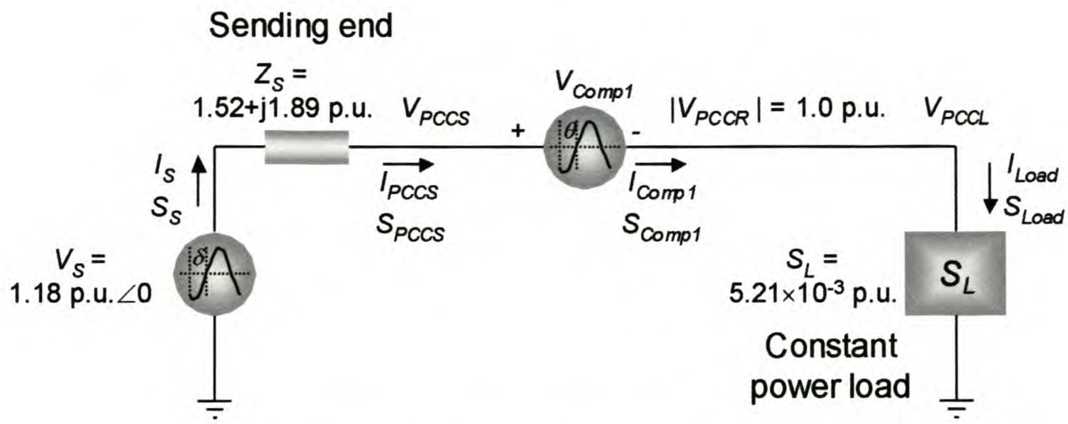


Fig. 4.24: Katima Mulilo network with a series compensator

The magnitude and angle across the series-compensated Katima Mulilo network are shown in Fig. 4.25. The voltage magnitude and angle injected by the series compensator varies as the voltage angle at the PCC is varied from $-\pi \leq \delta \leq \pi$ (between PCCS and PCCR). For different values of δ, the series compensator can be modelled as variable impedance connected in series with the line. By varying the line impedance with δ, the voltage drop across the line will not be constant for all values of δ, as shown in Fig. 4.25.

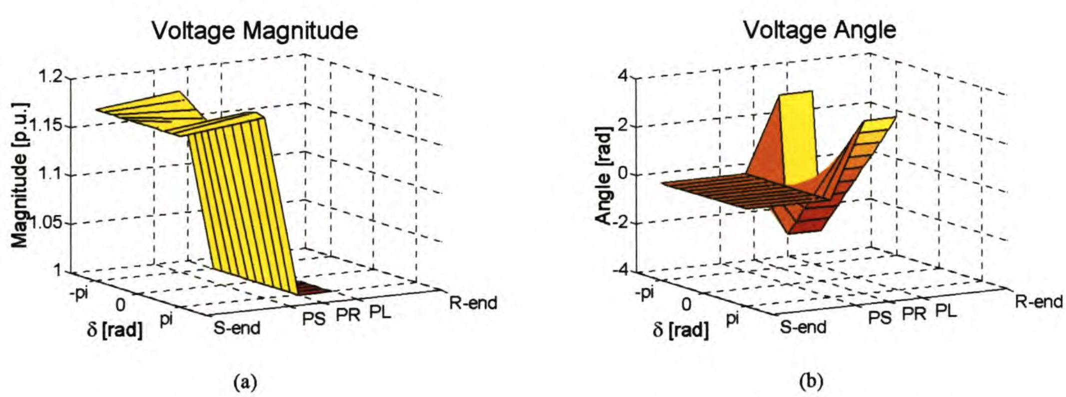
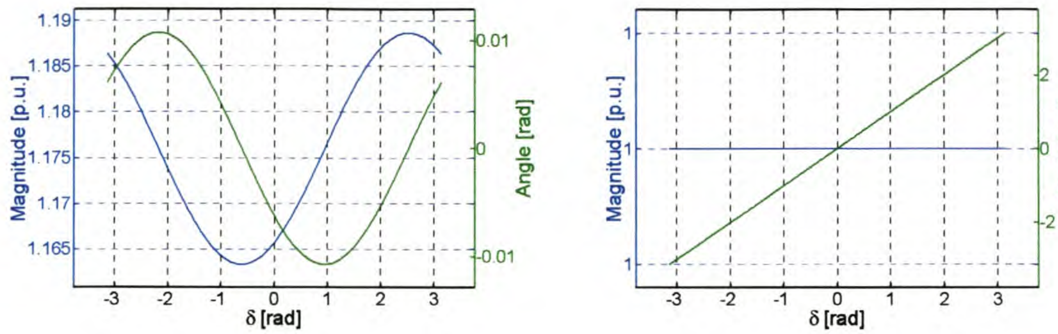


Fig. 4.25: Voltage magnitude and angle across the series-compensated Katima Mulilo network

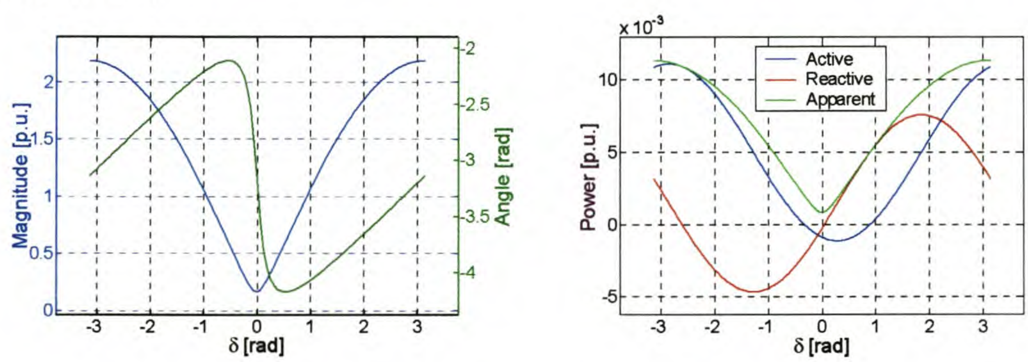
The voltage diagrams before and after the compensator at PCCS and PCCR are shown in Fig. 4.26 for an initial load of 5.21×10^{-3} p.u. The voltage regulation at PCCR is as desired, but varies between 1.16 and 1.19 p.u. at PCCS, which is acceptable as long as no loads are connected along the line and the line equipment is rated for overvoltages at PCCS.



(a) (b)

Fig. 4.26: Voltage plots at (a) PCCS and (b) PCCR

The true voltage added by the series compensator, as well as the power injected by the compensator, is shown Fig. 4.27. Although the series compensator must carry full load current, the voltage across the compensator is quite small (0.1658 p.u. for a minimum rating series compensator and 0.3779 p.u. for a purely reactive series compensator).



(a) (b)

Fig. 4.27: (a) Voltage and (b) power plots for the series compensator

The compensator has a minimum rating of 8.636×10^{-4} p.u. or 86.36 kVA with a power factor of –0.97 (83.77 kW and 20.99 kVar). This minimum rating is achieved at $\delta = 0$ rad. Compared to the minimum shunt voltage regulator, the series device rating is 97.93% less active power and 99.62% less apparent power. A SSSC equipped with energy storage can be used to supply both active and reactive power. As discussed previously, the generators at Katima Mulilo power station can be used as an energy source for this control scheme.

A purely reactive compensator can be implemented when the active power is zero. This occurs at $\delta = -0.32$ and $\delta = 0.8875$ rad. The former is chosen as this delivers the lowest compensator rating of -1.97×10^{-3} p.u. A series capacitor or a more complex device, such as a TCSC, with a rating of 197 kVar is required for this control strategy. Compared to purely reactive shunt compensation, the series device rating is 97.71% less reactive power, but requires additional bypass switches to prevent network fault currents from flowing through the compensator.

Transmission cost of the purely reactive series-compensated network

Because this is a series network the line current magnitude and angle stays constant throughout the network. The power losses of the series-compensated network will therefore be the same as for the uncompensated network discussed previously in section 4.2.3. The capital cost of the series capacitor and the equipment needed to bypass the capacitor during fault conditions just increases the LCC of the already expensive uncompensated AC network. For the series-compensated case the cost of losses cannot be used as outflows of capital in determination of the minimum present value of LCC, because the power losses are the same for both the uncompensated and the series-compensated network. The cost of the series capacitor and additional protection and control equipment were supplied by ABB and is shown in Table 4-11.

Table 4-11: Additional capital costs incurred for the series-compensated Katima Mulilo network

Description	Total [R]
197 kVar capacitor bank	2 100
Switchgear, civil work and steelwork	1 500 000
Primary over-voltage protection	1 200 000
Control and protection	210 000

Fig. 4.28 shows the present value of LCC for the lifetime of the uncompensated and series-compensated Katima Mulilo network and the results are summarised in Table 4-12.

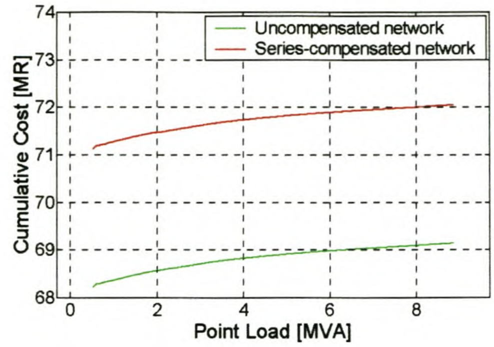


Fig. 4.28: LCC of the uncompensated and series-compensated Katima Mulilo network

The increased capital cost incurred with series compensation is clearly visible in Fig. 4.28. Due to the high cost of the equipment to protect the series capacitor against fault conditions, the purely reactive shunt compensator will be a more cost effective solution than the series compensator.

Table 4-12: Summary of the LCC of the uncompensated and series-compensated Katima Mulilo network

Network Configuration	PV_{LCC} [MR]		PV_{TCL} [MR]
	Minimum	Maximum	
Uncompensated	68.23	69.16	0.93
Series compensated	71.14	72.07	0.93

4.5 SERIES-SHUNT COMPENSATION

As its name indicates, the series-shunt compensator is a combination of the two devices. The series and shunt devices are fed from a common DC bus and, therefore, can exchange active power between them. The series-shunt device, therefore, has no need for an energy-storage device. With two devices, which can do both active and reactive power compensation, a potentially powerful and highly effective compensator is realized. Such a device not only has full four-quadrant operating capability in the complex $P + jQ$ plane, but also internal Var generation capability to support self-sufficiently the reactive power demand in any quadrant [B3].

With the network parameters from Fig. 4.12, the circuit diagram of the series-shunt compensated network is drawn in Fig. 4.29. Both the magnitude and the angle of V_{PCCR} can be controlled by the series part of the compensator. For consistency, $|V_{PCCR}|$ is specified to 1.0 p.u. θ_{PCCR} is also needed and specified as 0.1 rad. The load does not have a constant voltage or constant current characteristic, so θ_{PCCR} makes no difference to the load [C1]. The shunt compensator controls the power factor at the PCCS with reactive power injection. It is taken as unity to limit reactive power demand from Rundu substation at the sending end.

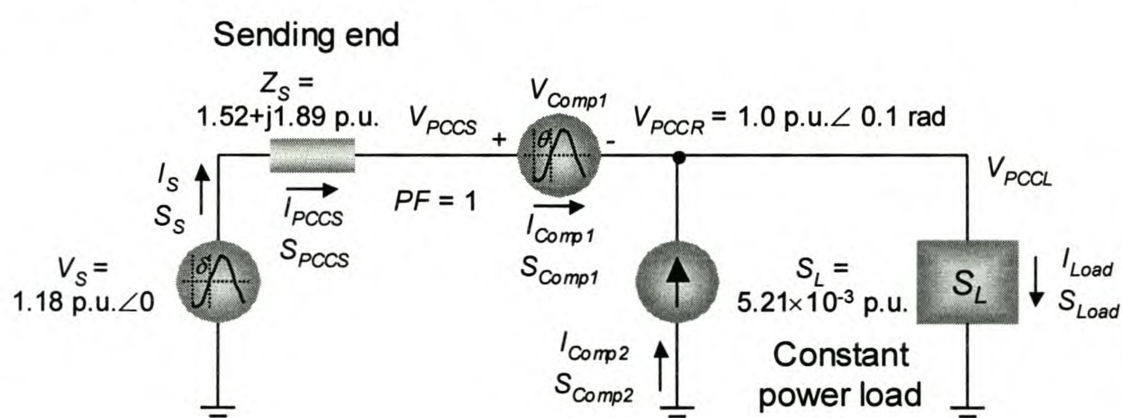
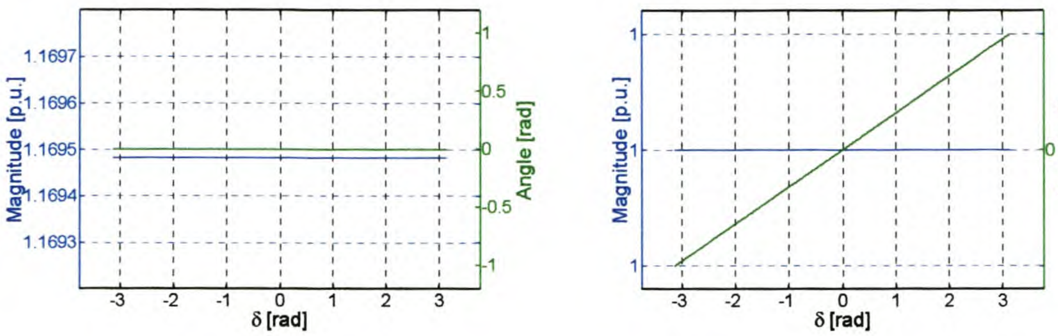


Fig. 4.29: Katima Mulilo network with a series-shunt compensator

The voltage diagrams generated by PSAT at the PCCS and PCCR are shown in Fig. 4.30 for an initial load of 5.21 x 10^{-3} p.u. The voltage regulation at the PCCR is the desired 1.0 p.u., but 1.1695 p.u. at the PCCS, which is well above NRS 048 specifications. No loads are connected along the line, but the line equipment must be rated for overvoltages at initial load conditions.



(a) (b)

Fig. 4.30: Voltage plots at (a) PCCS and (b) PCCR

The true voltage injected by the series compensator to regulate $|V_{PCCR}|$ is shown in Fig. 4.31. The compensator will have a maximum voltage rating of 2.17 p.u. at $\delta = 3.14$. Although the possibility exists that the compensator can increase the first swing stability margin, a compensated network will never be operated at such a high voltage angle. If a maximum voltage angle of 0.7 rad is assumed across the network, the maximum voltage rating will reduce to 0.77 p.u. The compensator will have a minimum rating of 0.17 at $\delta = 0$.

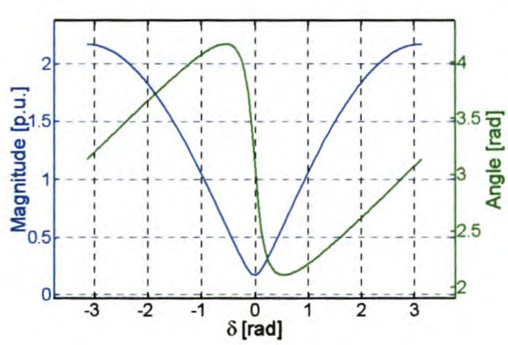


Fig. 4.31: True voltage injected by the series compensator

The active and reactive power delivery for the separate series and shunt compensators are shown in Fig. 4.32. The active power of the series and shunt devices are the opposite of one another, because the shunt compensator absorbs the active power required by the series compensator. This power is then transferred via the DC bus. DC bus losses are ignored; therefore, the series device injects all active power absorbed by the shunt device back into the network. The reactive power ratings differ because the series device regulates V_{PCCR}, while the shunt device regulates the power factor at PCCS.

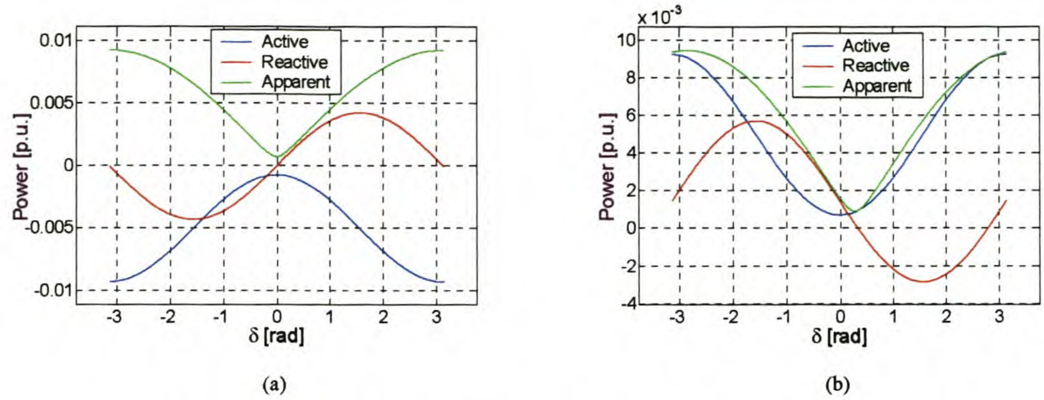


Fig. 4.32: Power requirements of the (a) series and (b) shunt devices

The maximum apparent power delivered by the series compensator is 3.282 x 10^{-3} p.u. (328.2 kVA) at $\delta = 0.7$. The maximum apparent power absorbed by the shunt compensator is 2.191 x 10^{-3} p.u. (219.1 kVA). The active power delivered and absorbed per compensator is 1.75 x 10^{-3} p.u. (175 kW) at this point. It can therefore be reasoned that the maximum rating of the series-shunt compensator is 547.3 kVA (350 kW).

The minimum apparent power delivered by the series compensator is 7.253 x 10^{-4} p.u. (72.53 kVA) at $\delta = 0$. The active power delivered and absorbed per compensator is 7.247 x 10^{-4} (72.47 kW) at this point. The minimum apparent power absorbed by the shunt compensator is 9.407 x 10^{-4} p.u. (94.07 kVA) at $\delta = 0.2513$, and the active power delivered and absorbed per compensator is 8.663 x 10^{-4} (86.63 kW).

Although the power rating of the series-shunt compensator is much less than the shunt device rating, the implementation of a series-shunt compensator for the Katima Mulilo network is technically impractical, due to the high voltage ratings of the series and shunt devices (102 kV and 132 kV). Such a device is economically not feasible and nor is it commercially available.

4.6 IN-LINE COMPENSATION – THE DC ALTERNATIVE

Medium-voltage direct current (MVDC) transmission can be used to bridge greater distances with low voltage and low power [A7]. MVDC can be considered as in-line compensation in the sense that it consists of a DC link with a voltage source converter (VSC) connected at either end. The advantages of using a VSC (with a series connection of IGBTs) for DC transmission include the following [A14]-[A17]:

- Independent control of active and reactive power
- Operation against isolated AC networks with no generation of their own
- Can operate at any short-circuit ratio
- Limited need of filters
- No converter transformers
- Can connect isolated generators to the grid or to other isolated loads.

For DC transmission the compensator at the sending end absorbs active and reactive power and transfers the active power across a common DC link to the receiving-end compensator. It, therefore, has no need for an energy-storage device. If the compensators are based on VSC technology, both active and reactive power can be transmitted in any direction. The reactive power capabilities of the converter can be used to control the AC voltages of the networks connected to the converter stations. With this type of operation the converter can be connected into an extremely weak AC network. The AC network can supply the reactive power, but capacitor banks are often used to supply this demand [A18]. The receiving-end compensator transmits the active power received over the DC link to the connected AC network. Since in a VSC the current can be forced to commutate, there is no need for a network to commutate against. In DC transmission it could then be very advantageous to use the VSC technology to supply passive or isolated networks such as Katima Mulilo.

4.6.1 DC line technology optimisation

The DC interface of TESAT is used to evaluate three different Nampower conductor choices in order to select the optimum conductor for a 500 km, ±50 kV bipolar DC line (shown in Fig. 4.1) between Rundu substation and Katima Mulilo distribution station. The following three ACSR conductors were evaluated with a standard 66 kV Nampower tower:

- Rabbit conductor
- Hare conductor
- Wolf conductor.

The line structure used by Nampower for the above-mentioned conductors are shown in Fig. 4.33. The data for the different conductors were sourced from ACSR property tables and are included in Addendum A.1.

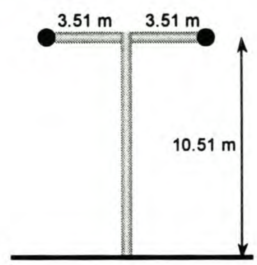


Fig. 4.33: 66 kV Nampower tower (Kamerad) [C18]

The DC resistance values at 40°C were calculated from the DC resistance values at 20°C. The resistance values for the different conductors are given in Table 4-13, as well as the deterministic ampacity values and the estimated cost of these lines.

Table 4-13: ±50 kV bipolar DC line data

Conductor	Tower	$r_{dc\ (40°C)}$ [Ω/km]	Ampacity [A]	Line Cost [R/km]
Rabbit	Kamerad	0.5863	196	68 500
Hare	Kamerad	0.2953	297	75 900
Wolf	Kamerad	0.1975	383	87 700

The data in Table 4-13 are entered in the DC interface of TESAT. A base value of 100 MW is used for S_{base} and 50 kV for V_{base}. The same values are used for interest rate, generation cost and ALF as for the AC line technologies. The additional cost of the converter stations, one at either end of the line, must be entered by the user, as well as the losses in each station as a percentage of the transmitted power. An estimated cost of R1.8 million per MW for the converter stations was used (R3 600 per MW-km), with the losses in each station estimated as 3% of the transmitted power. Details of the estimated converter station costs are included in

Addendum A.2. The cost per kilometre for the different DC lines was calculated in the same way as for the AC line technologies.

Voltage regulation analysis

From the TESAT analysis it was determined that the voltage regulation for the Rabbit and Hare conductors exceeds 40% for certain loads and therefore cannot be used to transmit the full load for the lifetime of the network (20 years). The voltage regulation of the ±50 kV bipolar Rabbit, Hare and Wolf lines is shown in Fig. 4.34 and the results are summarised in Table 4-14.

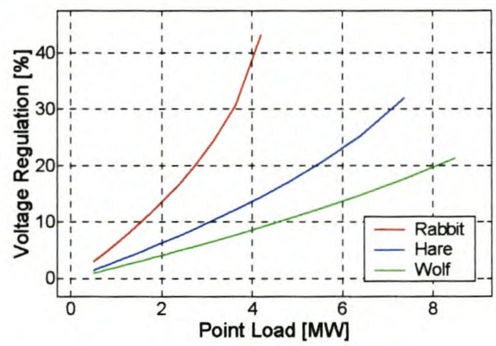


Fig. 4.34: Voltage regulation of the DC line technologies as a function of load growth

Fig. 4.34 shows that the Rabbit conductor will be able to transmit full load until the 15th year, while the Hare conductor will be able to transmit full load until the 19th year. Only the Wolf conductor will be able to transmit the full load for the lifetime of the line.

Table 4-14: Summary of voltage regulation performance

±50 kV	% Voltage Regulation	
Bipolar DC	Minimum	Maximum
Rabbit	3.0	43.2
Hare	1.5	32.1
Wolf	1.0	21.3

Transmission losses

The power losses in kW for the different bipolar DC lines are shown in Fig. 4.35 and the results are summarised in Table 4-15. These values include the power losses in the converter stations.

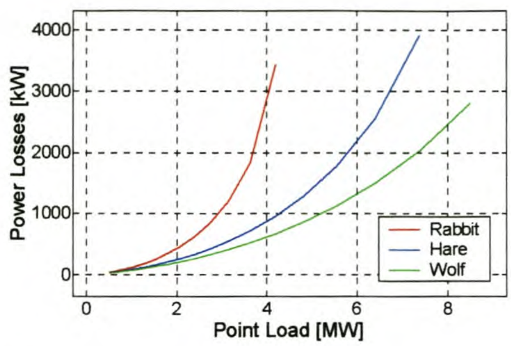


Fig. 4.35: Power losses of the DC line technologies as a function of load growth

The minimum values of power loss in each of the conductors include a 30 kW loss component for the two converter stations. The maximum value of converter losses for Rabbit and Hare conductors amounts to 251 kW and 443 kW respectively. The Wolf conductor will be the only conductor capable of transmitting the full load for the lifetime of the network, with maximum converter losses equal to 510 kW. The loss per pole ($P_{pole\ loss}$) for each of these lines can be calculated as follows:

$$\begin{aligned} P_{Losses} &= 2\cdot\left(P_{support\ loss} + P_{pole\ loss}\right) \\ \Rightarrow P_{pole\ loss} &= \frac{1}{2}P_{Losses} - P_{support\ loss} \end{aligned} \qquad (4\text{-}5)$$

The results of equation (4-5) are shown in Table 4-15 for each of the bipolar DC lines.

Table 4-15: Summary of power losses

±50 kV DC Line Technology	Minimum		Maximum	
	$P_{pole\ loss}$ [kW]	P_{Losses} [kW]	$P_{pole\ loss}$ [kW]	P_{Losses} [kW]
Rabbit	8	46	1 595	3 440
Hare	4	38	1 741	3 926
Wolf	3	35	1 153	2 816

Transmission costs

Fig. 4.36 shows the present value of LCC for the different bipolar DC lines and the results are summarised in Table 4-16. The capital cost and the cost of losses of the two converter stations are also included in the LCC.

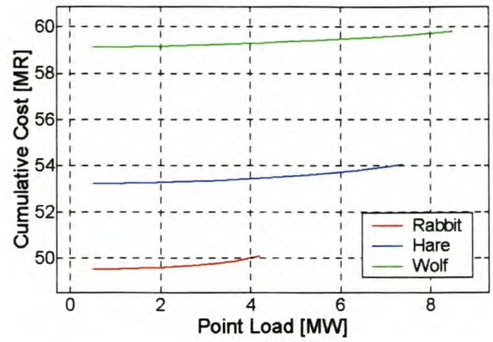


Fig. 4.36: LCC for the DC line technologies

From Fig. 4.36 it can be concluded that, although the Rabbit and Hare conductors are the most cost-effective choice to serve the intended load for most of the network lifetime, only the Wolf conductor will be able to transmit the full load for the lifetime of the network. The values of PV_{TCL} in Table 4-16 include both the cost of losses in the line and the two converter stations for the lifetime of the line.

Table 4-16: Summary of LCC and cost of losses

±50 kV DC Line Technology	PV_{LCC} [MR]		PV_{TCL} [MR]
	Minimum	Maximum	
Rabbit	49.55	50.11	0.56
Hare	53.25	54.05	0.80
Wolf	59.15	59.83	0.68

Appropriate technology index (ATI)

The KPIs to consider for DC line technologies are LCC, capacity and voltage regulation. As a result of the long distance and the low power requirements of the load, the DC voltage will be relatively high and the current low. Consequently, the capacity KPI of the line is ignored. For DC transmission the transistor-based inverter at the load end can cater for a voltage drop of more than 30%. The optimum line technology for DC transmission has to pursue a low-cost performance, which would economically justify low-power, long-distance transmission. The weighting factors for LCC and voltage regulation are, therefore, chosen as 80% and 20%, respectively.

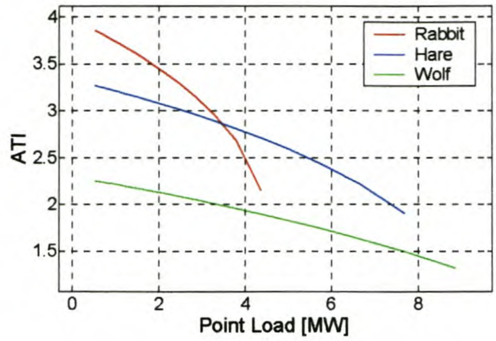


Fig. 4.37: ATI of the bipolar line technologies

The ATI for the bipolar line technologies is shown in Fig. 4.37 and the results are summarised in Table 4-17. Rabbit conductor would be the optimum conductor to string the line with for the first 13 years; after that Hare conductor would be the optimum choice until the 19th year. Only after 19 years does the Wolf conductor become the most effective choice in terms of cost and performance. However, owing to the capital cost incurred with conductor stringing, the Wolf conductor will be chosen as the optimum conductor for the network lifetime.

Table 4-17: ATI summary of the bipolar line technologies

±50 kV	ATI	
Bipolar DC	Minimum	Maximum
Rabbit	2.2	3.9
Hare	1.9	3.3
Wolf	1.3	2.3

4.6.2 Simplified network

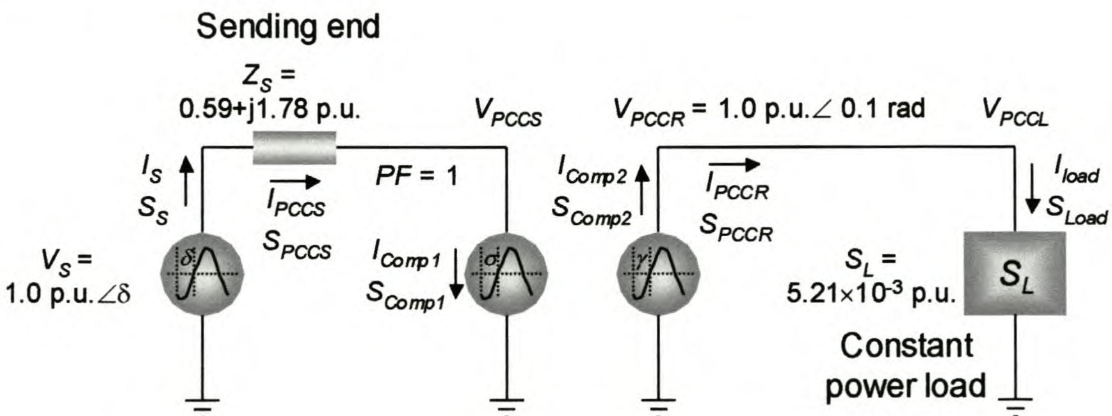
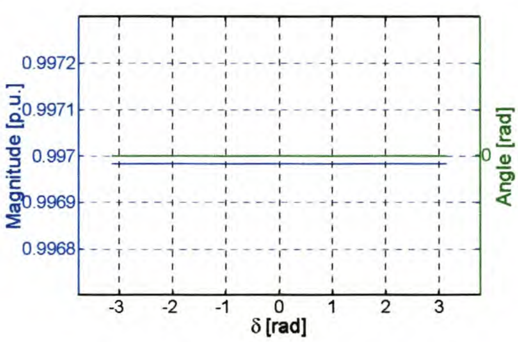


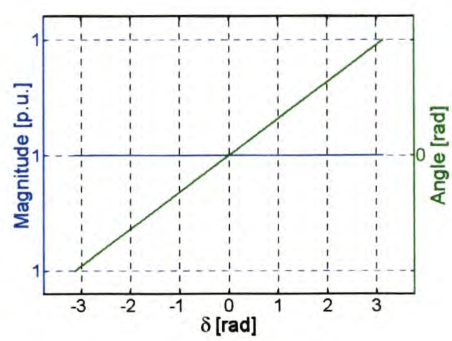
Fig. 4.38: Katima Mulilo network with an in-line compensator

The two compensators in Fig. 4.38, are connected via a 500 km, ±50 kV DC bipolar line. This equates to a maximum DC link voltage of 100 kV DC and from section 4.6.1 it was determined that the Wolf conductor would be the optimum conductor to use for this link.

The source impedance, Z_S, in Fig. 4.38 is equal to the p.u. impedance of the 66 kV busbar at Rundu substation, as shown in Table 4-6. $|V_{PCCR}|$ is specified as 1.0 p.u and θ_{PCCR} is specified as 0.1 rad. θ_{PCCR} only determines the load current angle, as the load has a fixed power factor of 0.96. The network planner must also specify the power factor at the PCCS, the DC line resistance and the operating voltage of the DC link. The power factor is taken as unity to limit reactive power demand from Rundu substation at the sending end. The DC line resistance is equal to 2.27 p.u. and the DC link voltage equals 1.52 p.u. As previously S_{base} = P_{base} = 100 MVA and V_{base} = 66 kV.



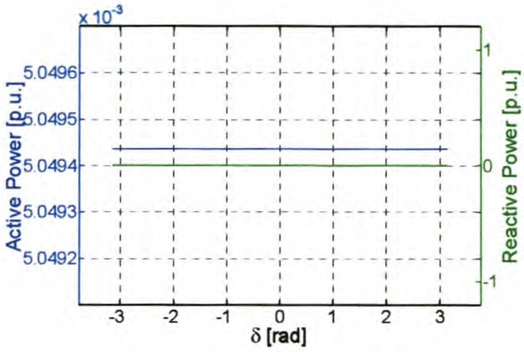
(a)



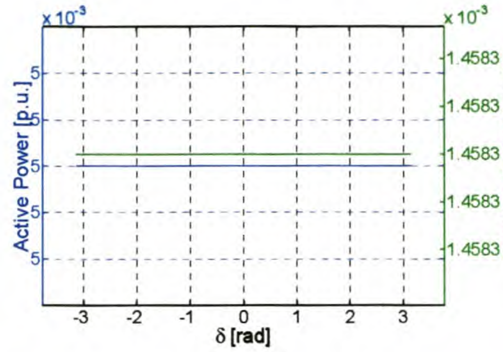
(b)

Fig. 4.39: Voltage plots at (a) PCCS and (b) PCCR for an initial load of 5.21 x 10^{-3} p.u.

The voltage diagrams before and after the compensator at PCCS and PCCR are shown in Fig. 4.39 for an initial load of 5.21 x 10^{-3} p.u. The voltage regulation at PCCR achieves the required 1.0 p.u., as well as the voltage regulation at the PCCS which is 0.9970 p.u.. This falls well within the NRS 048 voltage regulation specifications.



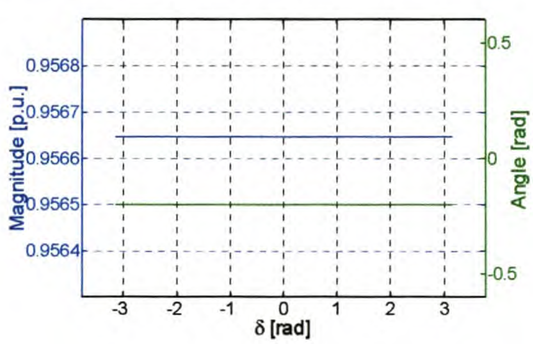
(a)



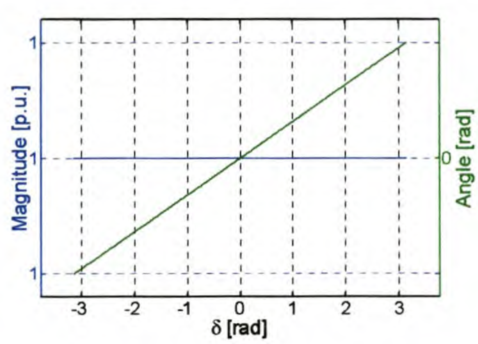
(b)

Fig. 4.40: Power requirements of the device at (a) PCCS and (b) PCCR for an initial load of 5.21×10^{-3} p.u.

With a power factor of unity, there is no reactive power delivered by the device at PCCS. As losses are included for the DC line, the active power injected at the PCCS will not be equal to the active power received over the DC link at the PCCR, as shown in Fig. 4.40. The DC line losses can be calculated from Fig. 4.40 (a) and (b) as 49.4×10^{-6} p.u., which translates into 4.94 kW. For a bipolar line the loss per pole equals 2.47 kW, which corresponds with the value calculated previously for DC Wolf conductor in Table 4-15. The device at the PCCR transmits the active power received over the DC link to the load. It also delivers 0.1458 MVar reactive power to the load for a 0.96 lagging power factor.



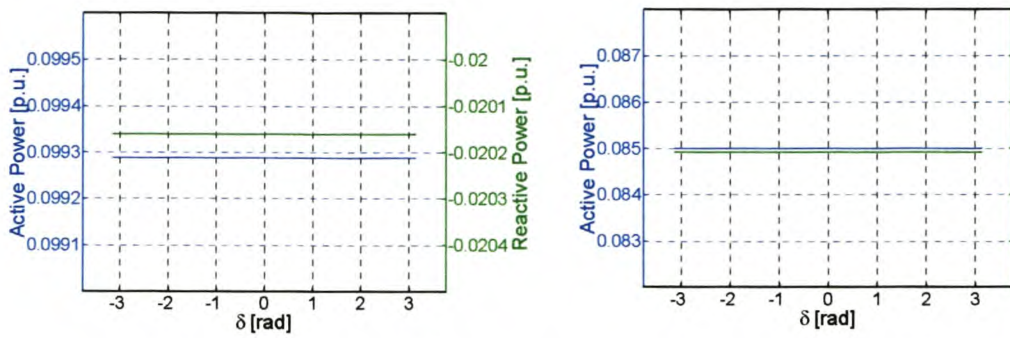
(a)



(b)

Fig. 4.41: Voltage plots at (a) PCCS and (b) PCCR for a final load of 88.54×10^{-3} p.u.

A power factor of –0.98 was chosen to achieve the desired voltage regulation at the PCCS for an anticipated final load of 88.54×10^{-3} p.u. The voltage regulation at the PCCR is regulated to the required 1.0 p.u., whereas the voltage regulation at the PCCS is 0.9566 p.u., which falls barely within the NRS 048 voltage regulation specification. The voltage graphs are shown in Fig. 4.41.



(a) (b)

Fig. 4.42: Power requirements of the device at (a) PCCS and (b) PCCR for a final load of 88.54×10^{-3} p.u.

With a leading power factor of 0.98, there is 2.02 MVar reactive power delivered by the device at the PCCS. The losses across the DC link can be calculated from Fig. 4.42 (a) and (b) as 0.0143 p.u., which translates into 1 430 kW. The load absorbs the power delivered by the device at the PCCR.

From Fig. 4.42 the sending-end compensator rating is 0.1013 p.u., which translates into 10.13 MVA and 9.93 MW at a power factor of 0.98 leading. A change in the power factor specified at the PCCS does not alter the active compensator rating, as power demanded by the load remains constant with V_{PCCR}. The receiving-end compensator rating is 0.0885 p.u., which translates into 8.85 MVA and 8.5 MW at a power factor of 0.96 lagging.

Determination of the break-even distance

The break-even distance between the uncompensated, shunt-compensated and series-compensated AC networks, and the DC (in-line compensated) network is shown in Fig. 4.43 (a) for final load conditions. It takes 255 km for DC transmission to break even with uncompensated AC transmission, 237 km to break even with shunt-compensated AC transmission and only 181 km to break even with series-compensated AC transmission for final load conditions.

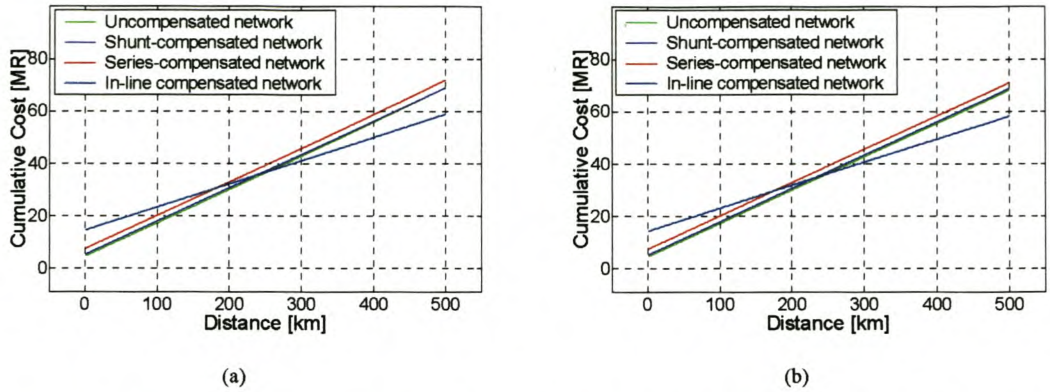


Fig. 4.43: Break-even distances for (a) final load conditions and (b) initial load conditions

For initial load conditions the break-even distance is reduced even further. The break-even distance is decreased to 250 km for uncompensated AC transmission, 232 km for shunt-compensated AC transmission and 178 km for series-compensated AC transmission. These curves are shown in Fig. 4.43 (b). Beyond these distances the in-line compensated network will become more economical than the uncompensated, shunt- and series-compensated networks.

Hingorani obtained similar results for the break-even distance in the very low power transmission range (5-20 MW) [A9].

4.7 SUMMARY OF THE KATIMA MULILO CASE STUDY

Summary of the line technologies

The Katima Mulilo case study served as a showcase for the practical application of the equations developed in Chapter 2 for three-phase AC and bipolar DC line technologies. A summary of the findings for the voltage regulation (VR), power losses, LCC and ATIs of different three-phase and bipolar line technologies are shown in Table 4-18.

Table 4-18: Summary of the line technology performance in the Katima Mulilo network

Line Technology			% VR		P_{Losses} [kW]		PV_{LCC} [MR]		ATI	
			Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.
132 kV	three-phase	Pelican	-9.8	-15.9	760	867	75.50	76.27	4.3	4.9
132 kV	three-phase	Twin Hare	-9.3	-15.6	1 307	1 522	89.50	90.85	3.6	4.1
132 kV	three-phase	Wolf	-6.8	-15.4	910	1 050	63.50	64.43	5.0	5.8
±50 kV	bipolar	Rabbit	3.0	43.2	46	3 440	49.55	50.11	2.2	3.9
±50 kV	bipolar	Hare	1.5	32.1	38	3 926	53.25	54.05	1.9	3.3
±50 kV	bipolar	Wolf	1.0	21.3	35	2 816	59.15	59.83	1.3	2.3

The three-phase lines suffer from voltage regulation problems. Due to the long distance and small point load, there is a rise in the receiving-end voltage, due to the flow of line-charging current through the line inductance. The AC line technologies will, therefore, require some form of line compensation. Since inductance and capacitance have no effect on a DC line, the voltage regulation problems experienced with the AC line technologies do not arise with DC transmission. No line compensation will be required to compensate for low receiving-end voltages, because the inverter at the load end can cater for a voltage drop of more than 30% with no commutation problems (transistor-based). For the DC line technologies only the Wolf conductor will be able to transmit the full load for the lifetime of the network.

The AC line technologies will suffer maximum power losses for initial load conditions, due to large line charging currents flowing through the line inductance for small loads over long distances. The maximum power losses for the DC line technologies are more than those for the AC line technologies, because the system voltage used for the AC technologies (132 kV phase-to-phase) is 1.52 times larger than the value that is used for the bipolar lines (±50 kV). Also, converter losses were included in the DC line losses.

Even though the cost of line compensation was not included in the LCC of the AC lines, the bipolar lines still prove to be a more economical solution than the more conventional three-phase AC lines. The Wolf conductor will be the cost effective solution for the AC line technologies. For the DC line technologies Wolf conductor will also be the most economic solution, because neither Rabbit nor Hare conductor will be able to transmit the full load for the lifetime of the network.

The ATIs for the line technologies are project specific with different upper and lower limits. The results are not absolute, but provide a quantative measure of the best option for a particular line technology, i.e. AC or DC. Results from the ATI analysis for the AC and DC lines also indicate Wolf conductor to be the optimum choice in both cases.

Summary of the support technologies

The Katima Mulilo network experiences voltage regulation problems for initial load conditions and the effect of shunt, series, series-shunt and in-line compensation in terms of cost and technical performance were considered for these networks. Table 4-19 is a summary of the findings for the voltage regulation of the Katima Mulilo network.

Table 4-19: Summary of the voltage regulator performance in the Katima Mulilo network

Compensator	Control	δ [rad]	$\lvert V_{PCCR} \rvert$ [p.u.]	Device Rating [MVA]	[MW]	[MVar]	Device Cost [MR]
None	None		1.17				None
Shunt	Purely reactive	0.10	1.00	8.60	0	8.60	0.70
Series	Purely reactive	-0.32	1.00	0.197	0	0.197	2.9121
Series-shunt	Minimum rating	0.70	1.00	0.547	0.350	0.420	None
In-line	Minimum rating	all	1.00	18.98	18.43	4.54	15.30

It was determined that Wolf conductor would be the optimum choice for a 500 km, 132 kV AC interconnection between Rundu substation and Katima Mulilo distribution station. The fault levels at Rundu substation are very low (high source impedance), which means that the connection of another 500 km AC line at this weak point is definitely out of the question. Compensation was, therefore, provided at Rundu substation to reduce the high source impedance, before an AC network was constructed between this point and Katima Mulilo. This compensation goes along with increased capital incurred for the AC network.

Without compensation at Katima Mulilo distribution station, the voltage rises to 1.17 p.u., which is well above the level of 1.05 required by NRS 048. All four compensator topologies solve the voltage-regulation problem. From the viewpoint of the AC network, there are two viable options, namely purely reactive shunt or series compensation.

Purely reactive shunt compensation requires 8.6 MVar reactive power absorption. The purely reactive series compensator has the lowest rating of all the options investigated. At 197 kVar it undercuts the rating of the purely reactive shunt compensator by 8.4 MVar. However, this is a very costly option, because the protection of the series device requires a very large capital outlay. Purely reactive shunt compensation is probably the most cost-effective solution to implement for the AC network, as a 7 MVar shunt reactor bank can be installed to absorb the bulk of the reactive power. The generators at Katima Mulilo power station can then be modified as synchronous condensers to rapidly compensate the balance of the required reactive power.

For this network a series-shunt device is definitely not appropriate. The ratings of the device are shown for 0.7 rad., because a network is never operated at voltage angles larger than this angle. Although the power ratings of the device are much less than purely reactive shunt or in-line compensation, the voltage ratings are excessive. For other networks these devices can be very effective.

DC transmission, as an in-line technology, has the highest rating of all the options investigated. One of the distinct advantages of DC transmission (transistor-based) for this case is that it can be connected into an extremely weak AC network. No compensation was therefore required at Rundu substation to reduce the source impedance, eliminating the extra capital costs incurred with compensation at this point. It was determined that Wolf conductor would be the optimum choice of a 500 km, ±50 kV bipolar DC line between Rundu substation and Katima Mulilo distribution station.

The capital outlay of the two converter stations required for DC transmission is the highest of all the compensator topologies. However, because the transmission distance is fairly large, the savings with the DC transmission line is enough to pay for the two converter stations. The in-line compensated network, therefore, seems to be the most effective solution in terms of cost. An economic analysis is necessary, which must include the cost of the AC and DC lines, terminals and any compensation needed for voltage support. The cost of losses must also be taken into account.

Summary of the economic analysis

Fig. 4.44 shows the present value of LCC for the different network solutions to the Katima Mulilo case study. A summary of the economic analysis is given in Table 4-20.

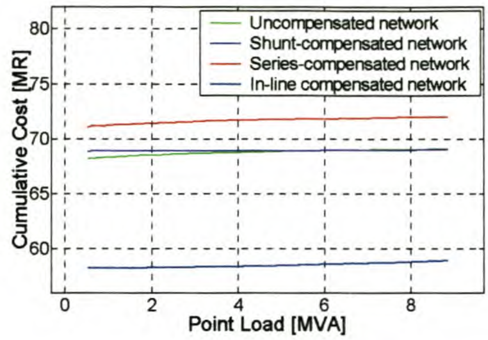


Fig. 4.44: PV_{LCC} for the different network solutions

From Fig. 4.44 and Table 4-20 it is proved that DC transmission will be the most cost-effective solution for the Katima Mulilo case study. A series-compensated AC network will be the most expensive solution, partly due to the cost of compensation required at Rundu substation and the high capital layout of the equipment to protect the series compensator against fault conditions. Although not clearly visible in the figure, the shunt-compensated AC network will be the second most economic solution for the network lifetime.

Table 4-20: Summary of the economic analysis for the Katima Mulilo case study

Network Configuration	PV_{LCC} [MR]		PV_{TCL} [MR]
	Minimum	Maximum	
Uncompensated	68.23	69.16	0.93
Shunt compensated	68.93	69.07	0.15
Series compensated	71.14	72.07	0.93
In-line compensated	58.30	58.98	0.68

The uncompensated and series-compensated networks have the highest cost of losses, because for light loading conditions these networks experience large power losses as a result of line charging currents producing more leading reactive power than the network can absorb. For shunt-compensated networks the purely reactive compensator absorbs this leading reactive power and, therefore, this network will have the lowest cost of losses.

Beyond this simplistic economic analysis there are many details, variations and performance issues that must be considered. Detailed studies should now be conducted to determine the best solution for the Katima Mulilo case study.

Summary of TESAT

TESAT made it possible to determine the optimum conductor for a particular AC or DC line technology within a matter of minutes. It is used as a first-assessment tool to evaluate the economic impact of different line and support technologies used in solving a network problem and to identify the most cost-effective solution. TESAT, together with PSAT, which was developed previously, will become useful tools for network planners, who need to optimise the use of line and support technologies to limit capital outlay and stretch networks up to thermal capacity. The innovative three-dimensional graphs provide additional insight into the voltage regulation, power losses, LCC and ATIs of different line technologies for various point load sizes and distances. Two-dimensional cuts enhance the understanding of the line technology performance even further.

CHAPTER 5 OTHER WORK AND FUTURE DEVELOPMENTS

5.1 PSAT – RETICMASTER INTERFACE

Provision is made in PSAT for two sources of network parameters. For a sub-transmission network the network parameters are calculated directly from the line parameters. Sub-transmission networks connect substations with substations or loads. The equivalent network is constructed from the known sending- or receiving-end voltage and the calculated line parameters. The calculation of the OLP as well as the interface with PSAT was discussed in detail in section 2.2.

Many loads are connected along the lines in reticulation networks. It was proven previously [C1] that the Thévenin equivalent network can be calculated for such a network, but is very tedious. The alternative is to obtain the electrical characteristics of the network from a ReticMaster model and use these to construct the sending- and receiving-end circuits. ReticMaster was developed by Inspired Interfaces to quickly and efficiently model radial networks from high-voltage source distribution down to individual low-voltage consumers [B18].

Fig. 5.1 portrays the sending end of the network, with many loads connected to the reticulation line. V_{SS}, S_{SS}, V_{PCCS}, and S_{PCCS} can be obtained from network simulation data at any two nodes in ReticMaster. With these parameters known, the lines and loads connected between these nodes can be represented by a single series and shunt impedance, as shown in (b). This method, however, is restricted to constant impedance loads between the nodes concerned. Since most of the loads in rural areas are constant impedances, this restriction is acceptable.

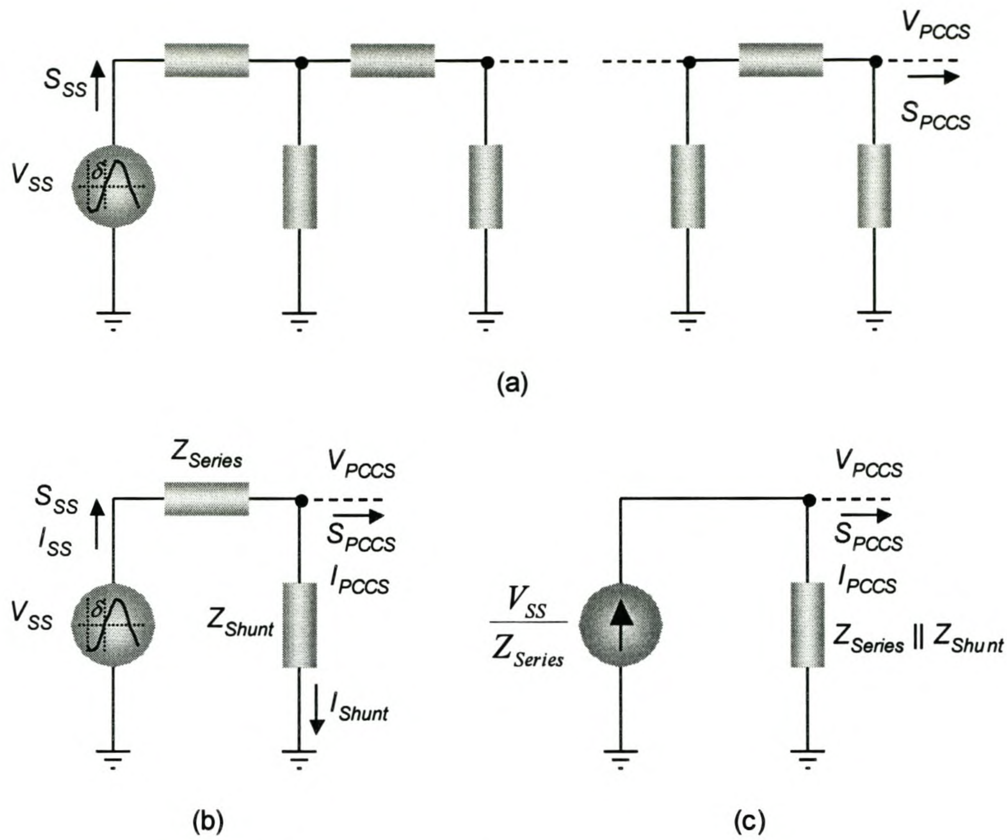


Fig. 5.1: Modelling of a reticulation network

From (b), Z_{Series}, and Z_{Shunt} are calculated as follows:

$$\begin{aligned} V_{Z_{Series}} &= V_{SS} - V_{PCCS} \\ I_{SS} &= \frac{S_{SS}^*}{V_{SS}^*} \\ Z_{Series} &= \frac{V_{Z_{Series}}}{I_{SS}} = \frac{|V_{SS}|^2 - V_{PCCS}V_{SS}^*}{S_{SS}^*} \end{aligned} \tag{5-1}$$

$$\begin{aligned} I_{PCCS} &= \frac{S_{PCCS}^*}{V_{PCCS}^*} \\ I_{Shunt} &= I_{SS} - I_{PCCS} \\ Z_{Shunt} &= \frac{V_{PCCS}}{I_{Shunt}} = \frac{|V_{PCCS}|^2 V_{SS}^*}{S_{SS}^* V_{PCCS}^* - S_{PCCS}^* V_{SS}^*} \end{aligned} \tag{5-2}$$

With V_{SS}, Z_{Series} and Z_{Shunt} known, a Thévenin-Norton source transformation results in (c). A Norton-Thévenin source transformation finally takes (c) to the sending end of PSAT, where V_S and Z_S are given by equations (5-3) and (5-4):

$$\begin{aligned} V_S &= \frac{V_{SS}}{Z_{Series}}\left(\frac{Z_{Series}Z_{Shunt}}{Z_{Series}+Z_{Shunt}}\right) \\ &= \frac{V_{SS}Z_{Shunt}}{Z_{Series}+Z_{Shunt}} \end{aligned} \tag{5-3}$$

$$\begin{aligned} Z_S &= Z_{Series} \parallel Z_{Shunt} \\ &= \frac{Z_{Series}Z_{Shunt}}{Z_{Series}+Z_{Shunt}} \end{aligned} \tag{5-4}$$

This method is not only restricted to the sending end of the network, but can also be applied to the receiving end.

5.1.1 Implementation

In this section the interface between PSAT and ReticMaster is discussed. Firstly a user report is created in ReticMaster, which is then exported as a text file. The electrical characteristics of the network are then extracted from this file by the PSAT-ReticMaster interface.

The user can define a report using the available fields by selecting *Tools/Options* from the main menu in ReticMaster and clicking on the *User Report Page* tab shown in Fig. 5.2. Multiple fields can be selected and moved to the *Report Fields* listbox. The order of how the fields appear in the report can also be changed using the move buttons [B18]. The fields that the PSAT-ReticMaster interface requires to calculate V and Z are displayed in the *Report Fields* listbox in Fig. 5.2. A file, *Report.rpt*, was created to ensure that the necessary fields would appear in the correct order in the *Report Fields* listbox. Selecting the *Retrieve a user report from a disk* button can retrieve this report.

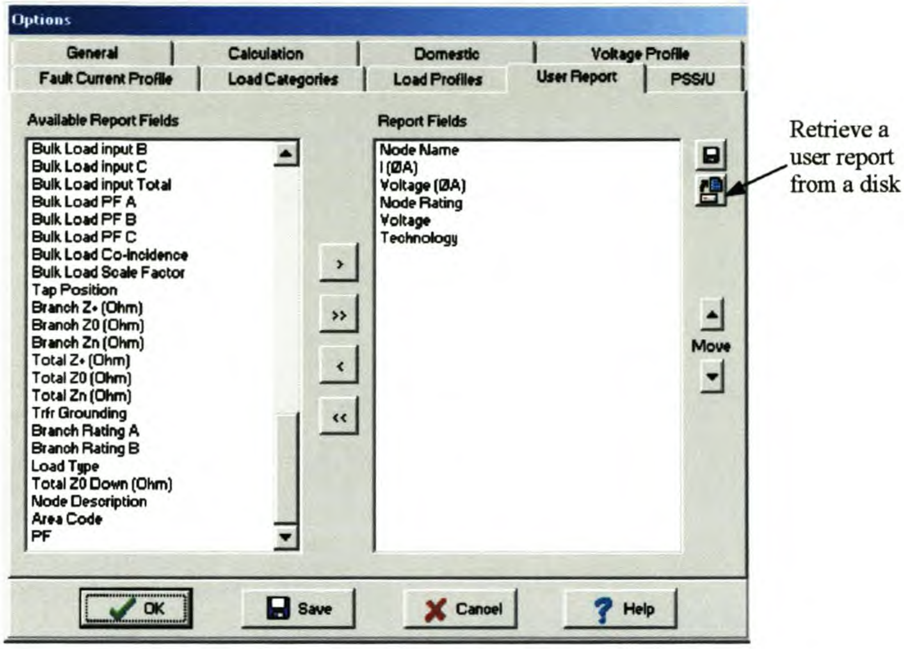


Fig. 5.2: ***User Report Page*** **tab in ReticMaster**

The user must specify the fields for the report before selecting *Tools/Report* from the main menu. A report can then be generated for either a portion of, or the complete, network. The user has the option of clicking on a required field and the report will be sorted according to this field. Buttons are available to select either ascending or descending order. Selecting *Saves the current file* button will export the user report to a text file for further analysis in the PSAT-ReticMaster interface.

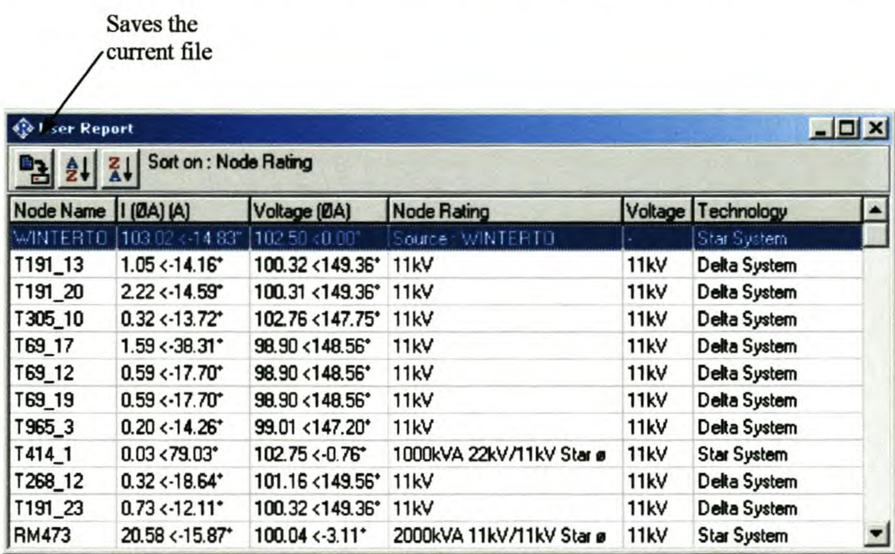


Fig. 5.3: The user report in ReticMaster

The interface to extract the electrical characteristics of a reticulation network from the ReticMaster user report is shown in Fig. 5.4. The user enters the ReticMaster report name, which is retrieved when the *Load* button is selected. The interface calculate V

and Z between the source node of the reticulation network and a user-selected load node if the *ReticMaster Source Node* radio button is selected. Different load nodes are selected by clicking on the node name in the *ReticMaster Load Node* listbox. If the *Alternative Source Node* radio button is selected, the interface will calculate V and Z between any two load or source nodes. $|V|$ and Z can be send to either the sending or the receiving end in the PSAT input interface.

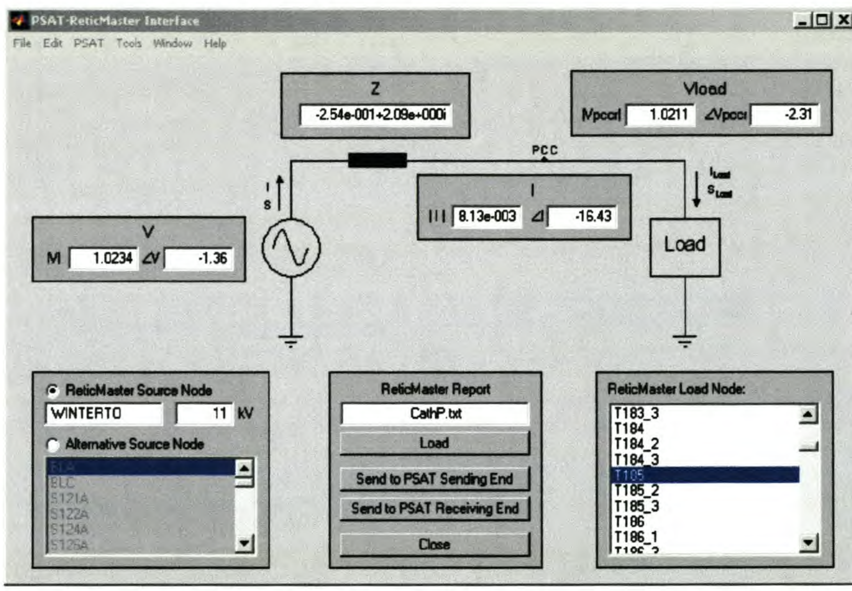


Fig. 5.4: Interface to extract network parameters from ReticMaster

Software development

The PSAT-ReticMaster interface was, just like the OLP interface of PSAT, implemented in Matlab. The reasons for choosing Matlab and the use of Matlab GUIs were discussed in section 2.2.4.

The equations to calculate V and Z are implemented in *interface.m*, of which the full program listing is available in Addendum B2.

ReticMaster version 4.2 was used to create the user report.

5.2 SUPPORT TECHNOLOGY AND DISPERSED GENERATION DATABASE

A database infrastructure has been developed to link the ratings and configurations determined by PSAT to real devices [C2]. Much data have been sourced and are currently fed into this library. The user interface of the database is shown in Fig. 5.5.

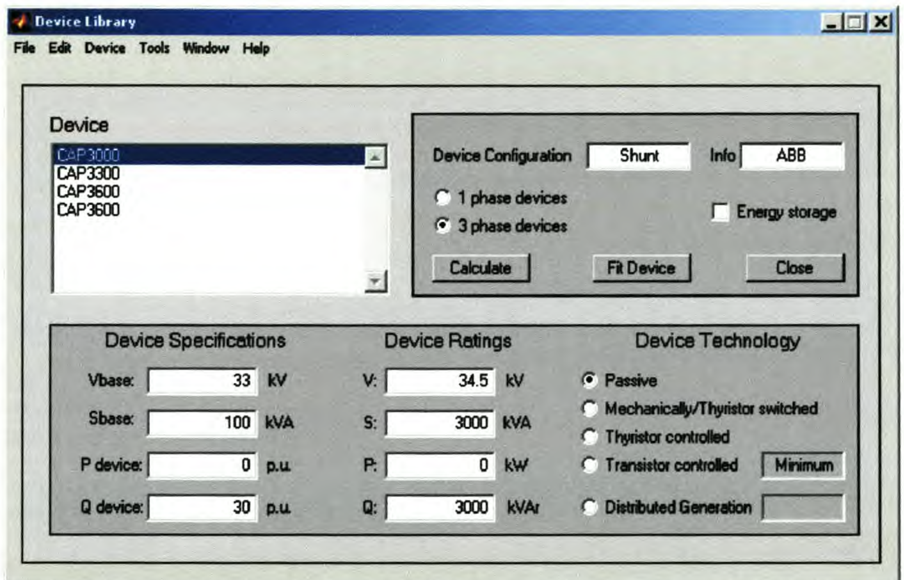


Fig. 5.5: Device library window

The following parameters are sent from PSAT to the device library:

- The device configuration – series, shunt, series-shunt or in-line;
- The value of P_{device} and Q_{device} in p.u. The device rating is determined by the specific network configuration and parameters, as well as the device configuration and control method;
- If the device requires energy storage, PSAT will mark the *Energy storage* checkbox. An energy-storage device can be connected to the DC bus of any transistor-controlled device [C19]. A device with energy storage or a distributed generator can supply both active and reactive power;
- For single-phase and three-phase networks PSAT enables the *1 phase devices* and *3 phase devices* radio buttons, respectively.

The user must enter the base values for V_{base} and S_{base}. PSAT determined the device configuration, but the user must decide the device technology. The network device technology will determine the reaction time and variation of the compensating power.

Depending on dynamic needs, the user must decide among the following technologies:

- Passive technology, such as shunt capacitors and reactors;
- Mechanically or thyristor switched technology, such as switched shunt capacitors and reactors or a tap-changing transformer;
- Thyristor-controlled technology, such as a Static Var Compensator (SVC), a TCSC or a HVDC line;
- Transistor-controlled technology, such as Static Synchronous Compensator (STATCOM) or a Dynamic Voltage Restorer (DVR).

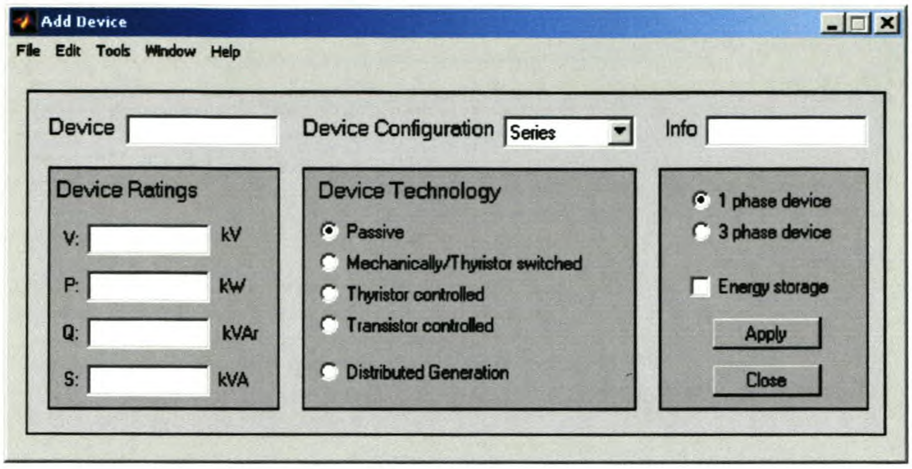
Distributed generation can be evaluated as a network support technology and is a special case of shunt compensation. It is important not to confuse this technology with a distributed generator as a power source at the receiving end of a network.

Selecting the *Calculate* button will automatically calculate the actual device ratings for a specified S_{base} and V_{base}. The values are displayed in the *Device Ratings* column of the interface. Pressing the *Fit Device* button will search the database for all similar devices with equal or larger ratings. The names of all the devices found will be displayed in the *Device* listbox. The ratings of these devices can be displayed in the *Device Ratings* column by clicking on the device names in the listbox. The *Info* box will display any additional data about the selected device, e.g. a manufacturer's name such as ABB. If no devices could be found for the specified device configuration, rating and technology, the message "no device" will be displayed in the *Info* box.

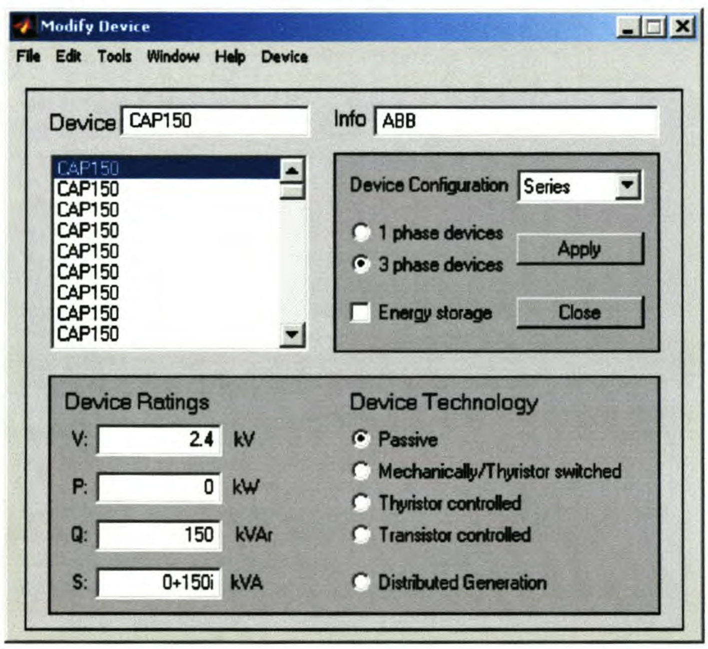
The user can add new devices to the database by selecting *Device/Add Device* from the main menu in the device library interface. Selecting *Device/Modify Device* from the main menu can modify existing devices. The add device and modify device interfaces of the device library are shown in Fig. 5.6.

Implementation

The device library interface was also implemented in Matlab. A discussion on the reasons for choosing Matlab and the use of Matlab GUIs was included in section 2.2.4.



(a)



(b)

Fig. 5.6: a) Add device and b) modify device interfaces

The program for the device library interface, as well as the add device and modify device interfaces, is implemented in *device.m*. The program starts by importing the device parameters send over by PSAT. It also imports the user-defined base values and device technologies. Next, all the devices in the database for the defined device technology and configuration are compared. All the devices found with equal or larger ratings are extracted from the database and listed in the device interface. The full program listing for *device.m* is included in Addendum B3.

5.3 FUTURE DEVELOPMENTS

Future developments in PSAT interfaces

The OLP interface of PSAT is a complete package for calculating the series impedance and shunt admittance for balanced three-phase, three-wire lines, where it is assumed the lines are perfectly transposed. The following future extensions are proposed for the OLP interface of PSAT:

- Extension of the conductor database to include other conductor types as well. These other conductor types include AAC, AAAC, aluminium conductor alloy-reinforced (ACAR), and aluminium-alloy conductor steel-reinforced (AACSR). Aerial bundled conductor (ABC) and steel conductor also need to be included in the database;
- The calculation of AC and DC resistance values at different operating temperatures for the various conductor types, as was discussed in detail in section 2.2.1;
- Equations were developed in sections 2.2.1 and 2.2.2 to calculate the phase and sequence impedances and admittances for a three-phase line with neutral conductors. This included the calculation of the series phase impedance matrix $\boldsymbol{Z_P}$ (2-31) and the shunt phase admittance matrix $\boldsymbol{Y_P}$ (2-54) for untransposed three-phase lines with neutral conductors. Equations were also developed to calculate the phase and sequence impedances and admittances of completely transposed lines. These equations can be implemented in the OLP interface to calculate the phase and sequence impedances of untransposed as well as transposed three-phase lines with and without neutral conductors. The phase and sequence values can also be calculated for other line technologies, such as phase-phase, single-phase or SWER, using the matrices developed in section 2.2. Because the effect of neutral wires is included, both balanced and unbalanced conditions can be considered.

These extensions will enable the OLP interface to accurately calculate the line parameters for any line technology with various conductor types, bundle selections and tower configurations.

Future developments in TESAT interfaces

It was shown extensively that TESAT could be used to do voltage regulation analysis and the calculation of line losses and transmission costs. Presently only three-phase lines are analysed in the AC interface of TESAT and bipolar lines in the DC interface. A technology index to compare different line technologies on a per project basis was also implemented in these interfaces. The objective of these technology indexes is to select the optimum conductor for a specific AC or DC line technology, based on the lowest transmission cost, best emergency performance, stability of transmission and quality of supply. The following future developments are proposed for the AC and DC interfaces of TESAT:

- Implementation of equations to calculate the voltage regulation, line losses and transmission costs of other line technologies, such as phase-to-phase, single-phase and SWER lines in the AC interface and monopolar lines in the DC interface of TESAT;
- The inclusion of these other line technologies in the AC and DC technology indexes, including the development of equations to calculate the emergency and stability performance of these line technologies;
- The development of equations to analyse the effect of network support devices on the voltage regulation, power losses and transmission costs of AC line technologies;
- The extension of the AC technology index to include the effect of line compensation. The analysis equations were documented in great detail in sections 2.6.1 and 2.7.1.

These extensions to the TESAT interfaces will enable network planners to determine the optimum line technology and conductor when designing new networks or upgrading existing networks. It will also allow planners to consider network support devices in order to operate lines closer to their thermal limits.

SWER as an alternative for low power, long distance transmission

The main benefit of SWER lines is that a single conductor is used, which means savings on line hardware. A SWER line can, therefore, be constructed at the fraction of the cost of conventional three-phase overhead lines. This section discusses the applicability of SWER networks for the Katima Mulilo case study. The phase and sequence impedances are calculated and the voltage regulation performance of the lines are analysed in TESAT.

The following two conductors are evaluated with a 132 kV Kamerad tower structure shown in Fig. 4.2:

- Wolf conductor
- Pelican conductor.

The AC resistance values at 40°C were calculated from the DC resistance values at 20°C and are shown in Table 4-1.

From section 2.2, the positive-sequence impedance for the SWER lines can be calculated as follows:

$$\begin{aligned}\hat{Z}_1 &= \hat{Z}_S - \hat{Z}_M \\ &= Z_{aaeq} \\ &= r_{aa} + r_{a'} + j\omega 2\times 10^{-4} \ln\frac{D_{aa'}}{D_{aa}} \\ &= 0.2585 + j0.7370 \quad [\Omega/\text{km}] \quad (\text{Wolf}) \\ &= 0.1784 + j0.7323 \quad [\Omega/\text{km}] \quad (\text{Pelican})\end{aligned} \tag{5-5}$$

and the shunt admittance as:

$$\begin{aligned}\hat{Y}_1 &= \hat{Y}_S - \hat{Y}_M \\ &= j\omega C_{aa} \\ &= j\omega\frac{2\pi\varepsilon}{\ln\left(D_{eq}/r\right)} \\ &= 2.25 \quad [\mu\text{S/km}] \quad (\text{Wolf}) \\ &= 2.29 \quad [\mu\text{S/km}] \quad (\text{Pelican})\end{aligned} \tag{5-6}$$

The values of series impedance and shunt capacitance for SWER lines will be larger than that for three-phase lines for the same choice of conductors. The reason for this is that the equivalent area perpendicular to the flux linkage is much larger.

The voltage regulation performance of the lines were analysed in TESAT for the Katima Mulilo network and the RXB values calculated in equations (5-5) and (5-6). A base value of 33⅓ MVA (100 MVA/3) is used for S_{base} and 76 kV ($132\ \text{kV}/\sqrt{3}$) for V_{base}. The results of the analysis are shown in Fig. 5.7.

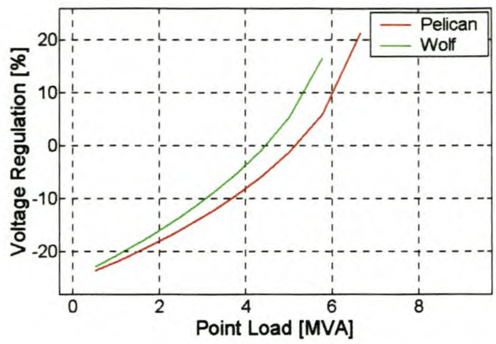


Fig. 5.7: Voltage regulation of 76.21 kV SWER lines as a function of load growth

The Wolf line will be able to conduct the full load until the 17th year and the Pelican line until the 18th year. Neither line will be able to conduct the full load for the lifetime of the network. Fig. 5.7 and Table 5-1 shows that for initial loading conditions high line-end voltages are experienced due to the Ferranti effect. Reactors will therefore be required for voltage control at the receiving end.

Table 5-1: Summary of the voltage regulation performance

76 kV AC SWER	% Voltage Regulation	
	Minimum	Maximum
Pelican	-23.57	21.37
Wolf	-22.77	16.55

The SWER networks can be upgraded to three-phase as the loading increases, allowing a phase implementation of infrastructure and delayed capital investment. For the Katima Mulilo case study it is, therefore, very advantageous to design the network to be extremely "light" initially with high flexibility by means of practical and cost effective upgrade paths to ensure that optimum use is made of resources and capital.

CHAPTER 6 CONCLUSION

In today's economic environment it is important to find the most effective solution in terms of cost and technical performance when planning new networks or upgrading existing networks. The application of line and support technologies in the design process can help achieve this goal. Network planners need tools to analyse and implement such line and support technologies. They also need to be trained in such analysis and implementation. An analysis tool, PSAT, has been developed previously to model various network configurations, network problems and support technologies as an aid to power system engineers for implementing network technologies [C1]. One of the objectives of this thesis was to document the development of a software tool, TESAT, to assist the network planner in selecting the optimum conductor and line technology for a network. If network support is required, i.e. to solve voltage regulation problems in an attempt to stretch the network up to thermal capacity, TESAT can be used to evaluate the economic impact of the line and support technology on network costs.

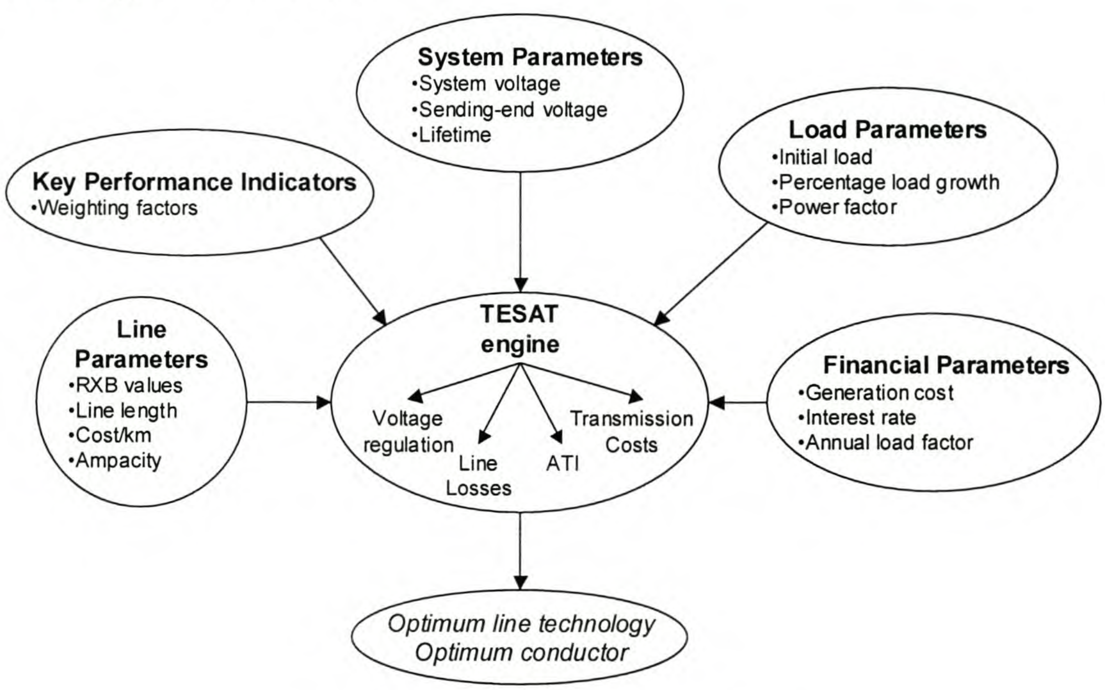


Fig. 6.1: Inputs and outputs of the TESAT engine

A diagram of the technical and financial inputs and outputs of the TESAT engine is shown in Fig. 6.1. The equations of the TESAT engine must consider the following inputs:

- Line parameters
- System parameters
- Load parameters

- Financial parameters
- Key performance indicators (KPIs).

The determination of the line parameters for three-phase AC and bipolar DC lines was discussed in Chapter 2. The equations for voltage regulation, transmission losses and LCC for these line technologies with different conductors were subsequently discussed. The cost of support technologies, if any, was included in the transmission costs of the line technologies.

Having determined the voltage regulation, losses and life cycle costs, the ATI was calculated to compare the various line technologies on a per project basis. For AC line technologies the KPIs to consider were voltage regulation, capacity, stability and voltage regulation. The DC ATI included the LCC, capacity and voltage regulation KPIs. The KPIs were summated using weighting factors to evaluate the overall technical and economical performance of a line technology.

The thermal load reach and transfer capabilities of different three-phase AC and bipolar DC lines at distribution voltage levels were determined in Chapter 3. The formula to calculate the load reach and transfer capabilities of these line technologies were derived from the voltage regulation equations developed in Chapter 2.

The Katima Mulilo case study was presented in Chapter 4 to show the practical application of TESAT. The case study showed that TESAT uses basic information to quickly calculate the voltage regulation, losses and LCC of different conductors from an AC and DC perspective in order to select the optimum conductor and line technology for the task at hand. The combination of network support technologies and line technologies to find the most effective solution in terms of cost and technical performance was also addressed by the case study. Ultimately, the case study showed that for low-power transmission over long distances, DC transmission at lower voltage levels is more economical than AC transmission at higher voltage levels. Apart from being technically more reliable than the AC interconnection, DC transmission will cost about MR 10 (15%) less than the most viable AC alternative for this particular case study.

Chapter 5 discussed the interfaces between PSAT and the real world. The interface between PSAT and ReticMaster was presented, as well as the development of the support technology and dispersed generation database. Future developments in the PSAT interfaces and TESAT were subsequently discussed.

TESAT provides a sturdy platform on which future developments such as the technical and economic performance of alternative line technologies can be modelled. However, TESAT can already operate as an analysis aid to determine the optimum conductor and line technology (three-phase or bipolar) for a network, and the economic impact of combining line and support technologies on network costs.

REFERENCES

Journals and conference papers

[A1] F.Engelbrecht, F.J. Rossouw and H.J. Beukes, *The Development of a Power System Analysis Tool*, 2001 SAUPEC, pp. 119-124, January 2001.

[A2] F.J. Rossouw and H.J. Beukes, *Analysis of Voltage Regulation and Network Support Technologies*, IASTED International Conference, pp. 594-602, September 2000.

[A3] V.T. Morgan, *Electrical Characteristics of Steel-cored Aluminium Conductors*, IEE Proceedings, Vol. 112, No. 2, pp. 325-334, February 1965.

[A4] C.F. Price and R.R. Gibbon, *Statistical Approach to Thermal Rating of Overhead Lines for Power Transmission and Distribution*, IEE Proceedings, Vol. 130, September 1983.

[A5] J. R. Carson, *Wave Propagation in Overhead Wires with Ground Return*, Bell System Technical Journal, Vol. 5., pp. 539-554, 1926.

[A6] C.W. Brice, *Voltage-Drop Calculations and Power-Flow Studies for Rural Electric Distribution Lines*, IEEE Transactions on Industry Applications, Vol. 28, No. 4, July/August 1992.

[A7] D. Povh, *Use of HVDC and Facts*, Proceedings of the IEEE, Vol. 88, No. 2, pp. 235-245, February 2000.

[A8] N.G. Hingorani, *High Voltage DC Transmission: A power electronics workhorse*, IEEE Spectrum, pp. 63-72, April 1996.

[A9] N.G. Hingorani, *DC Technology for Rural Transmission*, CIGRE international colloquium on HVDC and FACTS, paper 3.1, Johannesburg 1997.

[A10] B. Meyer and H.J. Beukes, Electronic Voltage Regulation of MV and LV Power Distribution Networks, Energize journal, pp. 24-26, November/December 2000.

[A11] Working Group 12 Cigre, *Probabilistic Determination of Conductor Current Ratings*, Electra Number 164, pp. 103 - 119, February 1996.

[A12] F. Engelbrecht and H.J. Beukes, *Analysis of AC and DC Transmission at Distribution Voltage Levels*, IASTED PES 2001, Rhodes, Greece, July 2001.

[A13] L.L. Dvorak, D.C. Dawson, R.W. Dempsey, H. Disante, C.M. Gadsden, J.D. III Huddleston and L.J. Schulze, *Summary of the 'Guide for the Protection of Shunt Reactors' ANSI C37.109*, IEEE Transactions on Power Delivery, pp. 116-118, January 1991.

[A14] G. Asplund, K. Eriksson, K. Svensson, *DC transmission based on Voltage Source Converters*, CIGRE SC14 Colloquium, Johannesburg, South Africa, 1997.

[A15] G. Asplund, K. Eriksson and B. Drugge, *Electric power transmission to distant loads by HVDC Light*, Distribution 2000, Sydney, Australia, 1997.

[A16] U. Axelsson, A. Holm, C. Liljegren, K. Eriksson and L. Weimers, *Gotland HVDC Light Transmission – World's First Commercial Small Scale DC Transmission*, CIRED Conference, Nice, France, May 1999.

[A17] R. Grünbaum, B. Halvarsson and A. Wilk-Wilczynski, *FACTS and HVDC Light for Power System Interconnections*, Power Delivery Conference, Madrid, Spain, September 1999.

[A18] A.E. Hammad and W.F. Long, *Performance and Economic Comparisons between Point-to-Point HVDC Transmission and Hybrid Back-to-Back HVDC/AC Transmission*, IEEE/PES 1989 Transmission and Distribution Conference, New Orleans, 1989.

Books and manuals

[B1] C. Adamson and N.G. Hingorani, *High Voltage Direct Current Power Transmission*, Garraway Ltd, London, 1960.

[B2] J. Arrillaga, *High Voltage Direct Current Transmission 2nd Edition*, IEE Power and Energy Series 29, England, 1998.

[B3] N.G. Hingorani and L. Gyugi, *Understanding FACTS*, IEEE Press, New York, 2000.

[B4] A.E. Guile, W. Paterson, *Electrical Power Systems (Second Edition)*, Pergamon Press, Oxford, 1977.

[B5] J.D. Glover and M. Sarma, *Power System Analysis and Design (Second Edition)*, PWS Publishing Company, Boston, 1994.

[B6] Westinghouse Electric Company, *Electrical Transmission and Distribution Handbook 4th Edition*, R.R. Donnelley and Sons Company, Chicago, 1964.

[B7] J.J. Grainger and W.D. Stevenson, *Power System Analysis*, McGraw-Hill Inc, 1994.

[B8] C.F. Wagner and R.D. Evans, *Symmetrical Components*, McGraw-Hill Inc., New York, 1933.

[B9] P.M. Anderson, *Analysis of Faulted Power Systems*, Iowa State Press, Ames, 1973.

[B10] Turan Gönen; *Modern Power System Analysis*, John Wiley and Sons Inc, New York, 1988.

[B11] P. Kundur; *Power System Stability and Control*, McGraw-Hill Inc., New York, 1994.

[B12] *Building GUIs with Matlab*, The Mathworks Inc., Natick M.A, 1997.

[B13] X. Wang and J.R. McDonald, *Modern Power System Planning*, McGraw-Hill Inc., Cambridge, 1994.

[B14] H. Lee Willis, *Power Distribution Planning Reference Book*, Marcel Dekker, Inc., New York, 1997.

[B15] Regulation 9 of Electricity Act, 1987 (as amended in January 1996).

[B16] NRS 048, Electricity Supply – Quality of Supply, Southern African Bureau of Standards, Pretoria, 1996.

[B17] T.J.E. Miller, *Reactive Power Control in Electric Systems*, John Wiley & Sons, New York, 1982.

[B18] *ReticMaster Quick Reference Guide*, Inspired Interfaces Ltd., Durban, South Africa, 2000.

Other references

[C1] F.J. Rossouw, *Analysis of Voltage Regulation and Network Support Technologies*, Masters thesis, University of Stellenbosch, October 2000.

[C2] H.J. Beukes, B. Meyer and F.J. Rossouw, *Network Options and Integration: Volume 3*, Eskom SAPSSI Research Report, Stellenbosch, October 2000.

[C3] C.F. Smit, *Calculation of Resistance of Overhead Line Conductors*, March 1996.

[C4] Aberdare Cables Africa Ltd, *Overhead Aluminium Conductor Tables*.

[C5] V.T. Morgan, *The Thermal Rating of Overhead Line Conductors, Part 1: The Steady State Thermal Model*, Electric Power Systems Research, pp. 119-139, 1982.

[C6] B. Meyer, *Opportunities for In-line, Transistor Based Technologies on MV and LV Power Distribution Networks*, Masters thesis, University of Stellenbosch, October 2000.

[C7] C.F. Smit, *Transmission Line Loadflow*, Draft document, February 2001.

[C8] M.C. Mpye, *Energy Loss Management Directive*, Eskom Energy Management Group, April 1999.

[C9] J. Jordaan, *Conductor Optimisation*, Eskom Transmission Line Technology Department, June 1996.

[C10] G. Stanford, *Distribution Standard: Part 6: Sub-transmission lines, Section 2: Conductors*, Eskom Distribution Technology Group, September 1999.

[C11] EED 15/6/1-1:1970, *Thermal Limits of Transmission Line and Busbar Conductors*, 1970.

[C12] C.F. Smit, *Conductor.xls*, Excel Spreadsheet.

[C13] G. Stanford, *Eskom Standard, Determination of Conductor Current Ratings in Eskom*, Eskom Distribution Technology Group, June 2000.

[C14] M. Manchen, F. Engelbrecht, Private discussions, Nampower Head Quarters in Windhoek, Namibia, February 2001.

[C15] G. Stanford, *Distribution Specification, Specification for Phase Conductor for Distribution Lines*, Eskom Distribution Technology Group, February 2001.

[C16] I.A. Ferguson, *Distribution Standard, Part 1: Planning guidelines, Section 42: Tool – Electrification technology selection*, Eskom Distribution Technology Group, February 2000.

[C17] EMCON Consulting Engineers, *Rural Electricity Distribution Master Plan for Namibia, Volume 2: Regional Planning Report for the Caprivi Region*, Internal Nampower Report, Namibia, August 2000.

[C18] Africa Grid Planning, *Power System Data Base*, Internal Nampower Report, Namibia, June 1999.

[C19] J.H.R. Enslin, H. du T. Mouton, D.D. Bester, A.D. le Roux and A.J. Visser, *Control and Implementation Considerations of a 2 MVA Power Quality Compensator*, Eskom SAPSSI Research Report, Stellenbosch, October 1999.

ADDENDUM A CONDUCTOR DATA AND CONVERTER STATION COSTS

A.1 CONDUCTOR DATA

Table A-1: Conductor Data [C4]

Conductor Name	Material Type	Stranding and Wire Diameter [mm]		Overall Diameter [mm]	Max DC Resistance [Ω/km, 20 °C]	DC Rating [A]
		Aluminium	Steel			
Ash	AAAC	19/3.48	—	17.40	0.1840	467
Chickadee	ACSR	18/3.77	1/3.77	18.87	0.1427	530
Fox	ACSR	6/2.79	1/2.79	8.37	0.7822	190
Gopher	ACSR	6/2.36	1/2.36	7.08	1.0933	150
Hare	ACSR	6/4.72	1/4.72	14.16	0.2733	360
Kingbird	ACSR	18/4.78	1/4.78	23.88	0.0894	
Magpie	ACSR	3/2.118	4/2.188	6.35	2.7070	92
Mink	ACSR	6/3.66	1/3.66	10.98	0.4546	260
Oak	AAAC	7/4.65	—	13.95	0.2790	359
Pelican	ACSR	18/4.14	1/4.14	20.70	0.1189	600
Rabbit	ACSR	6/3.35	1/3.35	10.05	0.5426	240
Squirrel	ACSR	6/2.11	1/2.11	6.33	1.3677	130
Tern	ACSR	45/3.38	7/2.25	27.00	0.0718	830
Wolf	ACSR	30/2.59	7/2.59	18.13	0.1828	470

A.2 ESTIMATED CONVERTER STATION COSTS

Each converter station consists of a voltage source converter, a DC bus, an AC filter and the converter control. The estimated converter station costs are shown in Table A-2.

Table A-2: Estimated converter station costs

Description	Quantity	Price/Unit [R]	Total [R]
Switches			
3.3 kV 400 A Dual Pack IGBTs	182	10 200.00	1 856 400.00
Heat Sinks			464 100.00
DC Bus			
DC Capacitors (3 300 μF, 450 V_{DC})	2 500	400.00	1 000 000.00
Filter Components			
Filter Capacitors (50 μF, 440 V_{AC})	16 434	130.00	2 136 420.00
Filter Inductors	6	100 000.00	600 000.00
Control and Measurement Equipment		2 000 000.00	2 000 000.00
Auxiliary Equipment		2 000 000.00	2 000 000.00
Sub-total 1			10 556 920.00
Other Costs			5 028 460.00
Total Cost			15 085 380.00
Total Cost/MW			1 774 751.00

The heart of each converter station is a three-phase voltage source converter with series connected IGBTs in each valve. For a maximum DC link voltage of 100 kV, each converter valve will consist of a minimum of 31 series connected 3.3 kV, 400 A dual pack IGBT switches. Every IGBT is provided with an antiparallel diode. Turn on/off of each IGBT is ordered via an optical link from the control equipment at ground potential. The semiconductors are cooled with heat sinks. The cost of these heat sinks was estimated as 25% of the total cost of the IGBT switches.

The primary objective of the DC capacitor bank is to provide a low inductive path for the turn-off current, and energy storage to be able to control the power flow [A14]. The DC capacitor bank consists of 10 parallel strings of 125 series connected 3 000 μF, 450 V electrolytic capacitors, in order to meet the voltage rating of 100 kV. Half of the DC bus capacitors are installed at the sending-end station and the other half at the receiving-end station. The DC bus will be robust against the effects of ripple current and the only other requirement is a reasonably stiff voltage on the DC side. Thus more parallel strings of series connected capacitors may be needed, although the DC line capacitance may be sufficient.

The converter generates characteristic harmonics related to the switching frequency. The AC currents are smoothed by the filter inductor and then the rest of the harmonic content of the AC voltage is reduced by the AC filter. The filter capacitors consist of 33 parallel strings of 83 series connected 50 μF, 440 V capacitors in order to meet the voltage rating of 66 kV phase-to-phase. The equivalent rating of the filter capacitors is 10 μF per phase, whereas the rating of the filter inductor depends on dynamic performance of the system and the ripple current limitations, which differs for each practical situation. The cost of the filter reactors was estimated as R100 000 each.

The cost of the DC bus LEMs is included in the cost for the control and measurement equipment. Auxiliary equipment costs include the cost of the DC line breakers, while other costs include transport costs, construction costs, field services, survey fees, engineering fees and general overhead costs. These were estimated as 50% of the total material cost.

For 8.5 MW long-distance transmission, with a ±50 kV bipolar DC line, the cost for the two converter stations would amount to R1.8 million per MW. It is important to note that the values given in Table A-2 are estimated values and are subject to change.

ADDENDUM B PROGRAM LISTINGS

B.1 OLP CODE

B.1.1 Conductor.m

```
function Conductor(action)
%Main Conductor GUI Function

switch(action)

case 'Apply_Settings'
	%Display Selected Conductor Parameters
	conductor=get(findobj(findobj('Tag','OLPGUI'),'Tag','CondSelect'),'UserData');
	value=get(findobj(findobj('Tag','OLPGUI'),'Tag','CondSelect'),'Value');
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Stranding'),'String',conductor(value).Stranding);
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Strand_Diam'),'String',conductor(value).StrandDiam);
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Overall_Diam'),'String',conductor(value).OverDiam);
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Rtext'),'String',conductor(value).R);
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Current'),'String',conductor(value).Current);

	%Frequency of Operation
	f=50;

	%Get Bundle Selection, Conductor Parameters, Tower Parameters, Distance and Earth Resistivity
	Bundleselect=get(findobj(findobj('Tag','OLPGUI'),'Tag','Conductors/Bundle'),'Value');
	StrDia=str2num(get(findobj(findobj('Tag','OLPGUI'),'Tag','Strand_Diam'),'String'));
	R=str2num(get(findobj(findobj('Tag','OLPGUI'),'Tag','Rtext'),'String'));
	Over_Diam=str2num(get(findobj(findobj('Tag','OLPGUI'),'Tag','Overall_Diam'),'String'));
	Distance=str2num(get(findobj(findobj('Tag','OLPGUI'),'Tag','Distance'),'String'));
	d=str2num(get(findobj(findobj('Tag','OLPGUI'),'Tag','Bundle Spacing'),'String'));
	p=str2num(get(findobj(findobj('Tag','OLPGUI'),'Tag','p'),'String'));
	Parameters=get(findobj('Tag','OLPGUI'),'UserData');

	GMD=Parameters(5);
	GMD0=Parameters(6);

	%Calculations

	%Calculation of the GMR of the conductor and the equivelent outside raduis (Dsc) of bundle selections
	x=conductor(value).x;
	if x==1.08955 %6/1
		axes(findobj(findobj('Tag','OLPGUI'),'Tag','Axes1'))
		cla;
		image(imread('ACSR6A1S.bmp','bmp'));
		set(gca,'Tag','Axes1');
		set(gca,'Visible','off');
		set(gca,'Position',[921 541 68 62]);
	elseif x==1.950552 %18/1
		axes(findobj(findobj('Tag','OLPGUI'),'Tag','Axes1'))
		cla;
		image(imread('ACSR18A1S.bmp','bmp'));
		set(gca,'Tag','Axes1');
		set(gca,'Visible','off');
		set(gca,'Position',[914 541 82 82]);
	elseif x==2.4118148 %24/7
		axes(findobj(findobj('Tag','OLPGUI'),'Tag','Axes1'))
		cla;
		image(imread('ACSR24A7S.bmp','bmp'));
		set(gca,'Tag','Axes1');
		set(gca,'Visible','off');
		set(gca,'Position',[900 522 110 112]);
	elseif x==2.56595 %26/7
		axes(findobj(findobj('Tag','OLPGUI'),'Tag','Axes1'))
		cla;
		image(imread('ACSR26A7S.bmp','bmp'));
		set(gca,'Tag','Axes1');
		set(gca,'Visible','off');
		set(gca,'Position',[900 521 111 111]);
	elseif x==2.892485 %30/7
		axes(findobj(findobj('Tag','OLPGUI'),'Tag','Axes1'))
		cla;
		image(imread('ACSR30A7S.bmp','bmp'));
		set(gca,'Tag','Axes1');
		set(gca,'Visible','off');
		set(gca,'Position',[897 520 115 115]);
	elseif x==3.174735 %45/7
		axes(findobj(findobj('Tag','OLPGUI'),'Tag','Axes1'))
		cla;
		%image(imread('ACSR45A7S.bmp','bmp'));
		set(gca,'Tag','Axes1');
		set(gca,'Visible','off');
		set(gca,'Position',[889 516 131 128]);
	elseif x==3.25136 %54/7
		axes(findobj(findobj('Tag','OLPGUI'),'Tag','Axes1'))
		cla;
		image(imread('ACSR54A7S.bmp','bmp'));
		set(gca,'Tag','Axes1');
		set(gca,'Visible','off');
		set(gca,'Position',[889 516 131 128]);
	else x==3.646384 %54/19
		axes(findobj(findobj('Tag','OLPGUI'),'Tag','Axes1'))
		cla;
		image(imread('ACSR54A19S.bmp','bmp'));
		set(gca,'Tag','Axes1');
		set(gca,'Visible','off');
		set(gca,'Position',[889 516 131 128]);
```

```
        end;

        Ds=(StrDia*x)/10;
        if Bundleselect==1
                GMR=Ds;
                Dsc=Over_Diam/20;
        elseif Bundleselect==2
                GMR=sqrt(Ds*d);
                Dsc=sqrt((Over_Diam/20)*d);
        elseif Bundleselect==3
                GMR=(Ds*d^2)^(1/3);
                Dsc=((Over_Diam/20)*d^2)^(1/3);
        elseif Bundleselect==4
                GMR=1.091*((Ds*d^3)^(1/4));
                Dsc=1.091*((Over_Diam/20)*d^3)^(1/4);
        end;

        %Calculation of R/km, L/km and C/km for positive and zero sequences
        R=R/Bundleselect;
        Re=9.869e-4*f;
        De=(2160*0.3048)*(sqrt(p/f));
        zs=(R+Re)+(2*pi*f*2e-4)*log(De/(GMR/100))*j;
        zm=Re+(2*pi*f*2e-4)*log(De/GMD)*j;
        z0=zs+2*zm;
        z1=zs-zm;

        R0=real(z0);
        R1=real(z1);
        L0=imag(z0)/(2*pi*f);
        L1=imag(z1)/(2*pi*f);

        GMR0=(Dsc/100)*(GMD^2);
        Cap1=(2*pi*8.854e-9)/(log(GMD/(Dsc/100)));
        Cap0=(2*pi*8.854e-9)/log(GMD0/GMR0);

        y1=j*2*pi*f*Cap1;
        y0=j*2*pi*f*Cap0;

        %Calculation of characteristic impedance and propagation constant
        Zc0=sqrt(z0/y0);
        Prop_Const0=sqrt(z0*y0);

        Zc1=sqrt(z1/y1);
        Prop_Const1=sqrt(z1*y1);

        %Calculation of ABCD parameters
        A0=cosh(Prop_Const0*Distance);
        D0=A0;
        B0=Zc0*sinh(Prop_Const0*Distance);
        C0=(1/Zc0)*sinh(Prop_Const0*Distance);

        A1=cosh(Prop_Const1*Distance);
        D1=A1;
        B1=Zc1*sinh(Prop_Const1*Distance);
        C1=(1/Zc1)*sinh(Prop_Const1*Distance);

        %Calculation of Z' and Y'/2
        Z1=B1;
        Y1=tanh((Prop_Const1*Distance)/2)/Zc1;

        Z0=B0;
        Y0=tanh((Prop_Const0*Distance)/2)/Zc0;

        %Store GMR, Distance, Dsc and StrDia in UserData field of MainDisplay figure
        Parameters(1)=StrDia;
        Parameters(2)=Dsc;
        Parameters(3)=Distance;
        Parameters(4)=GMR;
        Parameters(7)=R;
        Parameters(8)=p;

        %Outputs
        set(findobj(findobj('Tag','OLPGUI'),'Tag','GMRdisplay'),'String',sprintf('%3.4f',GMR));
        set(findobj(findobj('Tag','OLPGUI'),'Tag','z1edit'),'String',num2str(z1,'%1.4g'));
        set(findobj(findobj('Tag','OLPGUI'),'Tag','z0edit'),'String',num2str(z0,'%1.4g'));
        set(findobj(findobj('Tag','OLPGUI'),'Tag','y1edit'),'String',num2str(y1,'%1.3g'));
        set(findobj(findobj('Tag','OLPGUI'),'Tag','y0edit'),'String',num2str(y0,'%1.3g'));

        set(findobj(findobj('Tag','OLPGUI'),'Tag','Z1edit'),'String',num2str(Z1,'%1.4g'));
        set(findobj(findobj('Tag','OLPGUI'),'Tag','Y1edit'),'String',num2str(Y1,'%1.2e'));
        set(findobj(findobj('Tag','OLPGUI'),'Tag','Z0edit'),'String',num2str(Z0,'%1.4g'));
        set(findobj(findobj('Tag','OLPGUI'),'Tag','Y0edit'),'String',num2str(Y0,'%1.2e'));

        set(findobj('Tag','OLPGUI'),'UserData',Parameters);

case 'BundleSelect'
        value=get(findobj(findobj('Tag','OLPGUI'),'Tag','Conductors/Bundle'),'Value');

        if value==1
                set(findobj(findobj('Tag','OLPGUI'),'Tag','Bundletext'),'Enable','off');
                set(findobj(findobj('Tag','OLPGUI'),'Tag','Bundle Spacing'),'Enable','off');
                set(findobj(findobj('Tag','OLPGUI'),'Tag','Bundle Spacing'),'BackgroundColor',[.6941 .6941 .6941]);
                set(findobj(findobj('Tag','OLPGUI'),'Tag','Bundlecmtext'),'Enable','off');
        else
                set(findobj(findobj('Tag','OLPGUI'),'Tag','Bundletext'),'Enable','on');
                set(findobj(findobj('Tag','OLPGUI'),'Tag','Bundle Spacing'),'Enable','on');
                set(findobj(findobj('Tag','OLPGUI'),'Tag','Bundle Spacing'),'BackgroundColor',[1 1 1]);
                set(findobj(findobj('Tag','OLPGUI'),'Tag','Bundlecmtext'),'Enable','on');
        end;

case 'Modify'
```

```
        %Open Modify Conductor GUI with selected conductor parameters

        fig = findobj('Tag','Modify Conductor');
        if isempty(fig)
            open('ModConductor.fig');
        end

        conductor=get(findobj(findobj('Tag','OLPGUI'),'Tag','CondSelect'),'UserData');
        value=get(findobj(findobj('Tag','OLPGUI'),'Tag','CondSelect'),'Value');

        set(findobj('Tag','Name'),'String',conductor(value).Name);
        if conductor(value).x == 1.08955
            set(findobj(findobj('Tag','Modify Conductor'),'Tag','Strandingselect'),'Value',1);
            axes(findobj(findobj('Tag','Modify Conductor'),'Tag','Axes1'))
            cla;
            image(imread('ACSR6A1S.bmp','bmp'));
            set(gca,'Tag','Axes1');
            set(gca,'Color',[.8 .8 .8])
            set(gca,'Visible','off');
            set(gca,'Position',[56 61 68 62]);
        elseif conductor(value).x == 1.950552
            set(findobj(findobj('Tag','Modify Conductor'),'Tag','Strandingselect'),'Value',2);
            axes(findobj(findobj('Tag','Modify Conductor'),'Tag','Axes1'))
            cla;
            image(imread('ACSR18A1S.bmp','bmp'));
            set(gca,'Tag','Axes1');
            set(gca,'Color',[.8 .8 .8])
            set(gca,'Visible','off');
            set(gca,'Position',[51 46 82 82]);
        elseif conductor(value).x == 2.4118148
            set(findobj(findobj('Tag','Modify Conductor'),'Tag','Strandingselect'),'Value',3);
            axes(findobj(findobj('Tag','Modify Conductor'),'Tag','Axes1'))
            cla;
            image(imread('ACSR24A7S.bmp','bmp'));
            set(gca,'Tag','Axes1');
            set(gca,'Color',[.8 .8 .8])
            set(gca,'Visible','off');
            set(gca,'Position',[36 36 110 112]);
        elseif conductor(value).x == 2.56595
            set(findobj(findobj('Tag','Modify Conductor'),'Tag','Strandingselect'),'Value',4);
            axes(findobj(findobj('Tag','Modify Conductor'),'Tag','Axes1'))
            cla;
            image(imread('ACSR26A7S.bmp','bmp'));
            set(gca,'Tag','Axes1');
            set(gca,'Color',[.8 .8 .8])
            set(gca,'Visible','off');
            set(gca,'Position',[36 36 111 111]);
        elseif conductor(value).x == 2.892485
            set(findobj(findobj('Tag','Modify Conductor'),'Tag','Strandingselect'),'Value',5);
            axes(findobj(findobj('Tag','Modify Conductor'),'Tag','Axes1'))
            cla;
            image(imread('ACSR30A7S.bmp','bmp'));
            set(gca,'Tag','Axes1');
            set(gca,'Color',[.8 .8 .8])
            set(gca,'Visible','off');
            set(gca,'Position',[36 31 115 115]);
        elseif conductor(value).x == 3.174735
            set(findobj(findobj('Tag','Modify Conductor'),'Tag','Strandingselect'),'Value',6);
            axes(findobj(findobj('Tag','Modify Conductor'),'Tag','Axes1'))
            cla;
            %image(imread('ACSR??.bmp','bmp'));
            set(gca,'Tag','Axes1');
            set(gca,'Color',[.8 .8 .8])
            set(gca,'Visible','off');
            set(gca,'Position',[51 46 109 109]);
        elseif conductor(value).x == 3.25136
            set(findobj(findobj('Tag','Modify Conductor'),'Tag','Strandingselect'),'Value',7);
            axes(findobj(findobj('Tag','Modify Conductor'),'Tag','Axes1'))
            cla;
            image(imread('ACSR54A7S.bmp','bmp'));
            set(gca,'Tag','Axes1');
            set(gca,'Color',[.8 .8 .8])
            set(gca,'Visible','off');
            set(gca,'Position',[26 26 131 128]);
        else
            set(findobj(findobj('Tag','Modify Conductor'),'Tag','Strandingselect'),'Value',8);
            axes(findobj(findobj('Tag','Modify Conductor'),'Tag','Axes1'))
            cla;
            image(imread('ACSR54A19S.bmp','bmp'));
            set(gca,'Tag','Axes1');
            set(gca,'Color',[.8 .8 .8])
            set(gca,'Visible','off');
            set(gca,'Position',[26 26 131 128]);
        end;
        set(findobj(findobj('Tag','Modify Conductor'),'Tag','Strand_Diam'),'String',conductor(value).StrandDiam);
        set(findobj(findobj('Tag','Modify Conductor'),'Tag','Overall_Diam'),'String',conductor(value).OverDiam);
        set(findobj(findobj('Tag','Modify Conductor'),'Tag','Rtext'),'String',conductor(value).R);
        set(findobj(findobj('Tag','Modify Conductor'),'Tag','Current'),'String',conductor(value).Current);

    case 'ModStranding'
        Strandingselect=get(findobj(findobj('Tag','Modify Conductor'),'Tag','Strandingselect'),'Value');
        if Strandingselect==1
            axes(findobj(findobj('Tag','Modify Conductor'),'Tag','Axes1'))
            cla;
            image(imread('ACSR6A1S.bmp','bmp'));
            set(gca,'Tag','Axes1');
            set(gca,'Color',[.8 .8 .8])
            set(gca,'Visible','off');
            set(gca,'Position',[56 61 68 62]);
        elseif Strandingselect==2
            axes(findobj(findobj('Tag','Modify Conductor'),'Tag','Axes1'))
```

```
                cla;
                image(imread('ACSR18A1S.bmp','bmp'));
                set(gca,'Tag','Axes1');
                set(gca,'Color',[.8 .8 .8])
                set(gca,'Visible','off');
                set(gca,'Position',[51 46 82 82]);
        elseif Strandingselect==3
                axes(findobj(findobj('Tag','Modify Conductor'),'Tag','Axes1'))
                cla;
                image(imread('ACSR24A7S.bmp','bmp'));
                set(gca,'Tag','Axes1');
                set(gca,'Color',[.8 .8 .8])
                set(gca,'Visible','off');
                set(gca,'Position',[36 36 110 112]);
        elseif Strandingselect==4
                axes(findobj(findobj('Tag','Modify Conductor'),'Tag','Axes1'))
                cla;
                image(imread('ACSR26A7S.bmp','bmp'));
                set(gca,'Tag','Axes1');
                set(gca,'Color',[.8 .8 .8])
                set(gca,'Visible','off');
                set(gca,'Position',[36 36 111 111]);
        elseif Strandingselect==5
                axes(findobj(findobj('Tag','Modify Conductor'),'Tag','Axes1'))
                cla;
                image(imread('ACSR30A7S.bmp','bmp'));
                set(gca,'Tag','Axes1');
                set(gca,'Color',[.8 .8 .8])
                set(gca,'Visible','off');
                set(gca,'Position',[36 31 115 115]);
        elseif Strandingselect==6
                axes(findobj(findobj('Tag','Modify Conductor'),'Tag','Axes1'))
                cla;
                %image(imread('ACSR??.bmp','bmp'));
                set(gca,'Tag','Axes1');
                set(gca,'Color',[.8 .8 .8])
                set(gca,'Visible','off');
                set(gca,'Position',[51 46 109 109]);
        elseif Strandingselect==7
                axes(findobj(findobj('Tag','Modify Conductor'),'Tag','Axes1'))
                cla;
                image(imread('ACSR54A7S.bmp','bmp'));
                set(gca,'Tag','Axes1');
                set(gca,'Color',[.8 .8 .8])
                set(gca,'Visible','off');
                set(gca,'Position',[26 26 131 128]);
        else
                axes(findobj(findobj('Tag','Modify Conductor'),'Tag','Axes1'))
                cla;
                image(imread('ACSR54A19S.bmp','bmp'));
                set(gca,'Tag','Axes1');
                set(gca,'Color',[.8 .8 .8])
                set(gca,'Visible','off');
                set(gca,'Position',[26 26 131 128]);
        end;

case 'ApplyModConductor'

        %Get Conductor Structure
        %Get Changed Conductor Parameters

        conductor=get(findobj(findobj('Tag','OLPGUI'),'Tag','CondSelect'),'UserData');
        value=get(findobj(findobj('Tag','OLPGUI'),'Tag','CondSelect'),'Value');

        Name=get(findobj(findobj('Tag','Modify Conductor'),'Tag','Name'),'String');
        Strandingselect=get(findobj(findobj('Tag','Modify Conductor'),'Tag','Strandingselect'),'Value');
        Strand_Diam=str2num(get(findobj(findobj('Tag','Modify Conductor'),'Tag','Strand_Diam'),'String'));
        Overall_Diam=get(findobj(findobj('Tag','Modify Conductor'),'Tag','Overall_Diam'),'String');
        R=num2str(get(findobj(findobj('Tag','Modify Conductor'),'Tag','Rtext'),'String'));
        Current=num2str(get(findobj(findobj('Tag','Modify Conductor'),'Tag','Current'),'String'));

        Stranding=0;
        x=0;
        [Stranding,x]=GetStranding(Strandingselect);

        %Change Conductor Structure
        conductor(value)=struct('Name',Name,'Stranding',Stranding,'StrandDiam',Strand_Diam,'OverDiam',Overall_Diam,
                                        'R',R,'Current',Current,'x',x);

        for i=1:length(conductor)
                CondSelect(i)={conductor(i).Name};
        end

        %Outputs
        SetCondParameters(CondSelect,Stranding,Strand_Diam,Overall_Diam,R,Current,conductor);

case 'AddStranding'
        Strandingselect=get(findobj(findobj('Tag','Add Conductor'),'Tag','Strandingselect'),'Value');
        if Strandingselect==1
                axes(findobj(findobj('Tag','Add Conductor'),'Tag','Axes1'))
                cla;
                image(imread('ACSR6A1S.bmp','bmp'));
                set(gca,'Tag','Axes1');
                set(gca,'Color',[.8 .8 .8])
                set(gca,'Visible','off');
                set(gca,'Position',[56 61 68 62]);
        elseif Strandingselect==2
                axes(findobj(findobj('Tag','Add Conductor'),'Tag','Axes1'))
                cla;
                image(imread('ACSR18A1S.bmp','bmp'));
                set(gca,'Tag','Axes1');
```

```
                    set(gca,'Color',[.8 .8 .8])
                    set(gca,'Visible','off');
                    set(gca,'Position',[51 46 82 82]);
            elseif Strandingselect==3
                    axes(findobj(findobj('Tag','Add Conductor'),'Tag','Axes1'))
                    cla;
                    image(imread('ACSR24A7S.bmp','bmp'));
                    set(gca,'Tag','Axes1');
                    set(gca,'Color',[.8 .8 .8])
                    set(gca,'Visible','off');
                    set(gca,'Position',[36 36 110 112]);
            elseif Strandingselect==4
                    axes(findobj(findobj('Tag','Add Conductor'),'Tag','Axes1'))
                    cla;
                    image(imread('ACSR26A7S.bmp','bmp'));
                    set(gca,'Tag','Axes1');
                    set(gca,'Color',[.8 .8 .8])
                    set(gca,'Visible','off');
                    set(gca,'Position',[36 36 111 111]);
            elseif Strandingselect==5
                    axes(findobj(findobj('Tag','Add Conductor'),'Tag','Axes1'))
                    cla;
                    image(imread('ACSR30A7S.bmp','bmp'));
                    set(gca,'Tag','Axes1');
                    set(gca,'Color',[.8 .8 .8])
                    set(gca,'Visible','off');
                    set(gca,'Position',[36 31 115 115]);
            elseif Strandingselect==6
                    axes(findobj(findobj('Tag','Add Conductor'),'Tag','Axes1'))
                    cla;
                    %image(imread('ACSR??.bmp','bmp'));
                    set(gca,'Tag','Axes1');
                    set(gca,'Color',[.8 .8 .8])
                    set(gca,'Visible','off');
                    set(gca,'Position',[51 46 109 109]);
            elseif Strandingselect==7
                    axes(findobj(findobj('Tag','Add Conductor'),'Tag','Axes1'))
                    cla;
                    image(imread('ACSR54A7S.bmp','bmp'));
                    set(gca,'Tag','Axes1');
                    set(gca,'Color',[.8 .8 .8])
                    set(gca,'Visible','off');
                    set(gca,'Position',[26 26 131 128]);
            else
                    axes(findobj(findobj('Tag','Add Conductor'),'Tag','Axes1'))
                    cla;
                    image(imread('ACSR54A19S.bmp','bmp'));
                    set(gca,'Tag','Axes1');
                    set(gca,'Color',[.8 .8 .8])
                    set(gca,'Visible','off');
                    set(gca,'Position',[26 26 131 128]);
            end;

    case 'Add'

            %Get Conductor Structure
            %Get New Conductor Parameters
            conductor=get(findobj(findobj('Tag','OLPGUI'),'Tag','CondSelect'),'UserData');

            Name=get(findobj(findobj('Tag','Add Conductor'),'Tag','Name'),'String');
            Strandingselect=get(findobj(findobj('Tag','Add Conductor'),'Tag','Strandingselect'),'Value');
            Strand_Diam=str2num(get(findobj(findobj('Tag','Add Conductor'),'Tag','Strand_Diam'),'String'));
            Overall_Diam=get(findobj(findobj('Tag','Add Conductor'),'Tag','Overall_Diam'),'String');
            R=num2str(get(findobj(findobj('Tag','Add Conductor'),'Tag','Rtext'),'String'));
            Current=num2str(get(findobj(findobj('Tag','Add Conductor'),'Tag','Current'),'String'));

            Stranding=0;
            x=0;
            [Stranding,x]=GetStranding(Strandingselect)

            %Add New Conductor
            sz=size(conductor);

            conductor(sz(2)+1)=struct('Name',Name,'Stranding',Stranding,'StrandDiam',Strand_Diam,'OverDiam',Overall_Diam,'
                                            R',R,'Current',Current,'x',x);

            for i=1:length(conductor)
                    CondSelect(i)={conductor(i).Name};
            end

            %Outputs
            SetCondParameters(CondSelect,Stranding,Strand_Diam,Overall_Diam,R,Current,conductor);

    case 'OpenAddConductor'
            fig = findobj('Tag','Add Conductor');
            if isempty(fig)
                    open('AddCond.fig');
            end

    %Close Add Conductor GUI
    case 'CloseAddConductor'
            fig = findobj('Tag','Add Conductor');
            close(figure(fig));

    %Close Modify Conductor GUI
    case 'CloseModConductor'
            fig = findobj('Tag','Modify Conductor');
            close(figure(fig));

    case 'CloseOLP'
            fig = findobj('Tag','OLPGUI');
```

```
        saveas(figure(fig),get(fig,'FileName'),'fig');
        close(figure(fig));

end

%SubFunctions
function [Stranding,x]=GetStranding(Strandingselect)

        if Strandingselect==1
                Stranding='6/1';
                x=1.08955;
        elseif Strandingselect==2
                Stranding='18/1';
                x=1.950552;
        elseif Strandingselect==3
                Stranding='24/7';
                x=2.4118148;
        elseif Strandingselect==4
                Stranding='26/7';
                x=2.56595;
        elseif Strandingselect==5
                Stranding='30/7';
                x=2.892485;
        elseif Strandingselect==6
                Stranding='45/7';
                x=3.174735;
        elseif Strandingselect==7
                Stranding='54/7';
                x=3.25136;
        else
                Stranding='54/19';
                x=3.646384;
        end;

function SetCondParameters(CondSelect,Stranding,Strand_Diam,Overall_Diam,R,Current,conductor)
        set(findobj(findobj('Tag','OLPGUI'),'Tag','CondSelect'),'String',CondSelect);
        set(findobj(findobj('Tag','OLPGUI'),'Tag','Stranding'),'String',Stranding);
        set(findobj(findobj('Tag','OLPGUI'),'Tag','Strand_Diam'),'String',Strand_Diam);
        set(findobj(findobj('Tag','OLPGUI'),'Tag','Overall_Diam'),'String',Overall_Diam);
        set(findobj(findobj('Tag','OLPGUI'),'Tag','Rtext'),'String',R);
        set(findobj(findobj('Tag','OLPGUI'),'Tag','Current'),'String',Current);
        set(findobj(findobj('Tag','OLPGUI'),'Tag','CondSelect'),'UserData',conductor);
```

B.1.2 Tower.m

```
function Tower(action)
%Main Tower GUI Function

switch(action)

case 'TwoEarthWires'
	set(gcbo,'Value',1);
	set(findobj(findobj('Tag','OLPGUI'),'Tag','OneEarthRB'),'Value',0);
	set(findobj(findobj('Tag','OLPGUI'),'Tag','NoEarthRB'),'Value',0);
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Verte1'),'BackgroundColor',[1 1 1]);
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Horze1'),'BackgroundColor',[1 1 1]);
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Verte2'),'BackgroundColor',[1 1 1]);
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Horze2'),'BackgroundColor',[1 1 1]);
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Verte1'),'Enable','inactive');
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Horze1'),'Enable','inactive');
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Verte2'),'Enable','inactive');
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Horze2'),'Enable','inactive');
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Earth1mtext'),'Enable','on');
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Earth2mtext'),'Enable','on');
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Earth1text'),'Enable','on');
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Earth2text'),'Enable','on');

case 'OneEarthWire'
	set(gcbo,'Value',1);
	set(findobj(findobj('Tag','OLPGUI'),'Tag','TwoEarthRB'),'Value',0);
	set(findobj(findobj('Tag','OLPGUI'),'Tag','NoEarthRB'),'Value',0);
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Verte1'),'BackgroundColor',[1 1 1]);
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Horze1'),'BackgroundColor',[1 1 1]);
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Verte2'),'BackgroundColor',[.6941 .6941 .6941]);
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Horze2'),'BackgroundColor',[.6941 .6941 .6941]);
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Verte1'),'Enable','inactive');
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Horze1'),'Enable','inactive');
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Verte2'),'Enable','off');
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Horze2'),'Enable','off');
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Earth1mtext'),'Enable','on');
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Earth2mtext'),'Enable','off');
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Earth1text'),'Enable','on');
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Earth2text'),'Enable','off');


case 'NoEarthWires'
	set(gcbo,'Value',1);
	set(findobj(findobj('Tag','OLPGUI'),'Tag','TwoEarthRB'),'Value',0);
	set(findobj(findobj('Tag','OLPGUI'),'Tag','OneEarthRB'),'Value',0);
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Verte1'),'BackgroundColor',[.6941 .6941 .6941]);
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Horze1'),'BackgroundColor',[.6941 .6941 .6941]);
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Verte2'),'BackgroundColor',[.6941 .6941 .6941]);
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Horze2'),'BackgroundColor',[.6941 .6941 .6941]);
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Verte1'),'Enable','off');
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Horze1'),'Enable','off');
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Verte2'),'Enable','off');
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Horze2'),'Enable','off');
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Earth1mtext'),'Enable','off');
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Earth2mtext'),'Enable','off');
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Earth1text'),'Enable','off');
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Earth2text'),'Enable','off');

case 'Apply_Settings'
	%Set selected tower parameters
	tower=get(findobj(findobj('Tag','OLPGUI'),'Tag','TowerSelect'),'UserData');
	value=get(findobj(findobj('Tag','OLPGUI'),'Tag','TowerSelect'),'Value');

	set(findobj(findobj('Tag','OLPGUI'),'Tag','Horzaedit'),'String',tower(value).Horza);
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Horzbedit'),'String',tower(value).Horzb);
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Horzcedit'),'String',tower(value).Horzc);
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Horze1'),'String',tower(value).Horze1);
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Horze2'),'String',tower(value).Horze2);
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Vertaedit'),'String',tower(value).Verta);
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Vertbedit'),'String',tower(value).Vertb);
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Vertcedit'),'String',tower(value).Vertc);
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Verte1'),'String',tower(value).Verte1);
	set(findobj(findobj('Tag','OLPGUI'),'Tag','Verte2'),'String',tower(value).Verte2);

	%Frequency of operation
	f=50;

	%Get selected tower parameters
	Horza=str2num(get(findobj(findobj('Tag','OLPGUI'),'Tag','Horzaedit'),'String'));
	Horzb=str2num(get(findobj(findobj('Tag','OLPGUI'),'Tag','Horzbedit'),'String'));
	Horzc=str2num(get(findobj(findobj('Tag','OLPGUI'),'Tag','Horzcedit'),'String'));
	Horze1=str2num(get(findobj(findobj('Tag','OLPGUI'),'Tag','Horze1'),'String'));
	Horze2=str2num(get(findobj(findobj('Tag','OLPGUI'),'Tag','Horze2'),'String'));
	Verta=str2num(get(findobj(findobj('Tag','OLPGUI'),'Tag','Vertaedit'),'String'));
	Vertb=str2num(get(findobj(findobj('Tag','OLPGUI'),'Tag','Vertbedit'),'String'));
	Vertc=str2num(get(findobj(findobj('Tag','OLPGUI'),'Tag','Vertcedit'),'String'));
	Verte1=str2num(get(findobj(findobj('Tag','OLPGUI'),'Tag','Verte1'),'String'));
	Verte2=str2num(get(findobj(findobj('Tag','OLPGUI'),'Tag','Verte2'),'String'));

	Parameters=get(findobj('Tag','OLPGUI'),'UserData');

	GMR=Parameters(4);
	Distance=Parameters(3);
	Dsc=Parameters(2);
	R=Parameters(7);
	p=Parameters(8);

```

```
            SetRB(Horze1,Verte1,Horze2,Verte2)
            Name=tower(value).Name;
            PlotOLPTower(Name,Horza,Horzb,Horzc,Horze1,Horze2,Verta,Vertb,Vertc,Verte1,Verte2)

            %Calculations
            %Calculation of R/km, L/km and C/km for positive and zero sequences
            dab=sqrt((Verta-Vertb)^2+(Horza-Horzb)^2);
            dbc=sqrt((Vertb-Vertc)^2+(Horzb-Horzc)^2);
            dca=sqrt((Vertc-Verta)^2+(Horzc-Horza)^2);
            GMD=(dab*dbc*dca)^(1/3);

            dabi=sqrt((Verta+Vertb)^2+(Horza-Horzb)^2);
            daci=sqrt((Verta+Vertc)^2+(Horza-Horzc)^2);
            dbci=sqrt((Vertb+Vertc)^2+(Horzb-Horzc)^2);

            GMD0=2*(((Verta*Vertb*Vertc)*(dabi*daci*dbci)^2)^(1/9));
            GMR0=(Dsc/100)*(GMD^2);
            Cap0=(2*pi*8.854e-9)/log(GMD0/GMR0);
            Cap1=(2*pi*8.854e-9)/(log(GMD/(Dsc/100)));
            y0=j*2*pi*f*Cap0;
            y1=j*2*pi*f*Cap1;

            Re=9.869e-4*f;
            De=(2160*0.3048)*(sqrt(p/f));
            zs=(R+Re)+(2*pi*f*2e-4)*log(De/(GMR/100))*j;
            zm=Re+(2*pi*f*2e-4)*log(De/GMD)*j;
            z0=zs+2*zm;
            z1=zs-zm;

            R0=real(z0);
            R1=real(z1);
            L0=imag(z0)/(2*pi*f);
            L1=imag(z1)/(2*pi*f);

            %Calculation of characteristic impedance and propagation contstant
            Zc0=sqrt(z0/y0);
            Prop_Const0=sqrt(z0*y0);
            Zc1=sqrt(z1/y1);
            Prop_Const1=sqrt(z1*y1);

            %Calculation of ABCD parameters
            A0=cosh(Prop_Const0*Distance);
            D0=A0;
            B0=Zc0*sinh(Prop_Const0*Distance);
            C0=(1/Zc0)*sinh(Prop_Const0*Distance);

            A1=cosh(Prop_Const1*Distance);
            D1=A1;
            B1=Zc1*sinh(Prop_Const1*Distance);
            C1=(1/Zc1)*sinh(Prop_Const1*Distance);

            %Calculation of Z' and Y'/2
            Z1=B1;
            Y1=tanh((Prop_Const1*Distance)/2)/Zc1;

            Z0=B0;
            Y0=tanh((Prop_Const0*Distance)/2)/Zc0;

            Parameters(5)=GMD;
            Parameters(6)=GMD0;

            %Outputs
            set(findobj('Tag','OLPGUI'),'UserData',Parameters);
            set(findobj(findobj('Tag','OLPGUI'),'Tag','GMDdisplay'),'String',sprintf('%3.4f',GMD));
            set(findobj(findobj('Tag','OLPGUI'),'Tag','z1edit'),'String',num2str(z1,'%1.4g'));
            set(findobj(findobj('Tag','OLPGUI'),'Tag','z0edit'),'String',num2str(z0,'%1.4g'));
            set(findobj(findobj('Tag','OLPGUI'),'Tag','y1edit'),'String',num2str(y1,'%1.3g'));
            set(findobj(findobj('Tag','OLPGUI'),'Tag','y0edit'),'String',num2str(y0,'%1.3g'));
            set(findobj(findobj('Tag','OLPGUI'),'Tag','Z1edit'),'String',num2str(Z1,'%1.4g'));
            set(findobj(findobj('Tag','OLPGUI'),'Tag','Y1edit'),'String',num2str(Y1,'%1.2e'));
            set(findobj(findobj('Tag','OLPGUI'),'Tag','Z0edit'),'String',num2str(Z0,'%1.4g'));
            set(findobj(findobj('Tag','OLPGUI'),'Tag','Y0edit'),'String',num2str(Y0,'%1.2e'));


    case 'Add'
            %Get Tower Structure

            tower=get(findobj(findobj('Tag','OLPGUI'),'Tag','TowerSelect'),'UserData');

            %Get New Tower Parameters
            Name=get(findobj(findobj('Tag','Add Tower'),'Tag','AddName'),'String');
            Horza=str2num(get(findobj(findobj('Tag','Add Tower'),'Tag','AddHorzaedit'),'String'));
            Horzb=str2num(get(findobj(findobj('Tag','Add Tower'),'Tag','AddHorzbedit'),'String'));
            Horzc=str2num(get(findobj(findobj('Tag','Add Tower'),'Tag','AddHorzcedit'),'String'));
            Verta=str2num(get(findobj(findobj('Tag','Add Tower'),'Tag','AddVertaedit'),'String'));
            Vertb=str2num(get(findobj(findobj('Tag','Add Tower'),'Tag','AddVertbedit'),'String'));
            Vertc=str2num(get(findobj(findobj('Tag','Add Tower'),'Tag','AddVertcedit'),'String'));
            Verte2=str2num(get(findobj(findobj('Tag','Add Tower'),'Tag','AddVerte2'),'String'));
            Verte1=str2num(get(findobj(findobj('Tag','Add Tower'),'Tag','AddVerte1'),'String'));
            Horze2=str2num(get(findobj(findobj('Tag','Add Tower'),'Tag','AddHorze2'),'String'));
            Horze1=str2num(get(findobj(findobj('Tag','Add Tower'),'Tag','AddHorze1'),'String'));

            %Add New Tower Structure
            sz=size(tower);

            tower(sz(2)+1)=struct('Name',Name,'Horza',Horza,'Horzb',Horzb,'Horzc',Horzc,'Horze1',Horze1,'Horze2',Horze2,
                                  'Verta',Verta,'Vertb',Vertb,'Vertc',Vertc,'Verte1',Verte1,'Verte2',Verte2);

            for i=1:length(tower)
                    TowerSelect(i)={tower(i).Name};
            end
```

```

		%Outputs
		PlotOLPTower(Name,Horza,Horzb,Horzc,Horze1,Horze2,Verta,Vertb,Vertc,Verte1,Verte2);
		PlotAddGUITower(Horza,Horzb,Horzc,Horze1,Horze2,Verta,Vertb,Vertc,Verte1,Verte2)
		SetRB(Horze1,Verte1,Horze2,Verte2)
		SetTowerParameters(TowerSelect,Horza,Horzb,Horzc,Horze1,Horze2,Verta,Vertb,Vertc,Verte1,Verte2,tower);

	case 'Modify'
		%Open Modify Tower GUI with selected tower parameters

		fig = findobj('Tag','Modify Tower');
		if isempty(fig)
			open('ModTower.fig');
		end

		tower=get(findobj(findobj('Tag','OLPGUI'),'Tag','TowerSelect'),'UserData');
		value=get(findobj(findobj('Tag','OLPGUI'),'Tag','TowerSelect'),'Value');

		set(findobj(findobj('Tag','Modify Tower'),'Tag','ModName'),'String',tower(value).Name);
		set(findobj(findobj('Tag','Modify Tower'),'Tag','ModHorzaedit'),'String',tower(value).Horza);
		set(findobj(findobj('Tag','Modify Tower'),'Tag','ModHorzbedit'),'String',tower(value).Horzb);
		set(findobj(findobj('Tag','Modify Tower'),'Tag','ModHorzcedit'),'String',tower(value).Horzc);
		set(findobj(findobj('Tag','Modify Tower'),'Tag','ModVertaedit'),'String',tower(value).Verta);
		set(findobj(findobj('Tag','Modify Tower'),'Tag','ModVertbedit'),'String',tower(value).Vertb);
		set(findobj(findobj('Tag','Modify Tower'),'Tag','ModVertcedit'),'String',tower(value).Vertc);
		set(findobj(findobj('Tag','Modify Tower'),'Tag','ModVerte2'),'String',tower(value).Verte2);
		set(findobj(findobj('Tag','Modify Tower'),'Tag','ModVerte1'),'String',tower(value).Verte1);
		set(findobj(findobj('Tag','Modify Tower'),'Tag','ModHorze2'),'String',tower(value).Horze2);
		set(findobj(findobj('Tag','Modify Tower'),'Tag','ModHorze1'),'String',tower(value).Horze1);

		Horza=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModHorzaedit'),'String'));
		Horzb=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModHorzbedit'),'String'));
		Horzc=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModHorzcedit'),'String'));
		Verta=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModVertaedit'),'String'));
		Vertb=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModVertbedit'),'String'));
		Vertc=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModVertcedit'),'String'));
		Verte2=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModVerte2'),'String'));
		Verte1=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModVerte1'),'String'));
		Horze2=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModHorze2'),'String'));
		Horze1=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModHorze1'),'String'));

		PlotModGUITower(Horza,Horzb,Horzc,Horze1,Horze2,Verta,Vertb,Vertc,Verte1,Verte2)

	case 'ApplyModTower'
		%Get Tower Structure
		tower=get(findobj(findobj('Tag','OLPGUI'),'Tag','TowerSelect'),'UserData');
		value=get(findobj(findobj('Tag','OLPGUI'),'Tag','TowerSelect'),'Value');

		%Get Modified Tower Parameters
		Name=get(findobj(findobj('Tag','Modify Tower'),'Tag','ModName'),'String');
		Horza=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModHorzaedit'),'String'));
		Horzb=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModHorzbedit'),'String'));
		Horzc=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModHorzcedit'),'String'));
		Verta=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModVertaedit'),'String'));
		Vertb=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModVertbedit'),'String'));
		Vertc=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModVertcedit'),'String'));
		Verte2=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModVerte2'),'String'));
		Verte1=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModVerte1'),'String'));
		Horze2=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModHorze2'),'String'));
		Horze1=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModHorze1'),'String'));


		tower(value)=struct('Name',Name,'Horza',Horza,'Horzb',Horzb,'Horzc',Horzc,'Horze1',Horze1,'Horze2',Horze2,
						'Verta',Verta,'Vertb',Vertb,'Vertc',Vertc,'Verte1',Verte1,'Verte2',Verte2);

		for i=1:length(tower)
			TowerSelect(i)={tower(i).Name};
		end

		%Outputs
		PlotOLPTower(Name,Horza,Horzb,Horzc,Horze1,Horze2,Verta,Vertb,Vertc,Verte1,Verte2)
		PlotModGUITower(Horza,Horzb,Horzc,Horze1,Horze2,Verta,Vertb,Vertc,Verte1,Verte2)
		SetRB(Horze1,Verte1,Horze2,Verte2)
		SetTowerParameters(TowerSelect,Horza,Horzb,Horzc,Horze1,Horze2,Verta,Vertb,Vertc,Verte1,Verte2,tower);

	case 'OpenAddTower'
		fig = findobj('Tag','Add Tower');
		if isempty(fig)
			open('AddTower.fig');
		end

	%Close Add Tower GUI
	case 'CloseAddTower'
		fig = findobj('Tag','Add Tower');
		close(figure(fig));

	%Close Modify Tower GUI
	case 'CloseModTower'
		fig = findobj('Tag','Modify Tower');
		close(figure(fig));

	end

	%SubFunction
	function SetTowerParameters(TowerSelect,Horza,Horzb,Horzc,Horze1,Horze2,Verta,Vertb,Vertc,Verte1,Verte2,tower);
		set(findobj(findobj('Tag','OLPGUI'),'Tag','TowerSelect'),'String',TowerSelect);
		set(findobj(findobj('Tag','OLPGUI'),'Tag','TowerSelect'),'UserData',tower);
		set(findobj(findobj('Tag','OLPGUI'),'Tag','Horzaedit'),'String',Horza);
		set(findobj(findobj('Tag','OLPGUI'),'Tag','Horzbedit'),'String',Horzb);
		set(findobj(findobj('Tag','OLPGUI'),'Tag','Horzcedit'),'String',Horzc);
		set(findobj(findobj('Tag','OLPGUI'),'Tag','Horze1'),'String',Horze1);
```

```
        set(findobj(findobj('Tag','OLPGUI'),'Tag','Horze2'),'String',Horze2);
        set(findobj(findobj('Tag','OLPGUI'),'Tag','Vertaedit'),'String',Verta);
        set(findobj(findobj('Tag','OLPGUI'),'Tag','Vertbedit'),'String',Vertb);
        set(findobj(findobj('Tag','OLPGUI'),'Tag','Vertcedit'),'String',Vertc);
        set(findobj(findobj('Tag','OLPGUI'),'Tag','Verte1'),'String',Verte1);
        set(findobj(findobj('Tag','OLPGUI'),'Tag','Verte2'),'String',Verte2);
        set(findobj(findobj('Tag','OLPGUI'),'Tag','TowerSelect'),'UserData',tower);

function PlotOLPTower(Name,Horza,Horzb,Horzc,Horze1,Horze2,Verta,Vertb,Vertc,Verte1,Verte2)
        oneearth=get(findobj(findobj('Tag','OLPGUI'),'Tag','OneEarthRB'),'Value');
        twoearth=get(findobj(findobj('Tag','OLPGUI'),'Tag','TwoEarthRB'),'Value');
        xmax=0;
        ymax=0;

        axes(findobj(findobj('Tag','OLPGUI'),'Tag','Axes2'))
        cla;
        plot(Horza,Verta, 'r o');
        hold on;
        set(gca,'Tag','Axes2');
        if Horza>xmax
                xmax=Horza;
        end;
        if Vertc>ymax
                ymax=Verta;
        end;

        xmax=Horza;
        ymax=Verta;
        set(gca,'Tag','Axes2');
        plot(Horzb,Vertb,       'w o');
        hold on;
        set(gca,'Tag','Axes2');
        if Horzb>xmax
                xmax=Horzb;
        end;
        if Vertb>ymax
                ymax=Vertb;
        end;

        plot(Horzc,Vertc, 'b o');
        hold on;
        set(gca,'Tag','Axes2');
        if Horzc>xmax
                xmax=Horzc;
        end;
        if Vertc>ymax
                ymax=Vertc;
        end;

        if (oneearth==1) & (isempty(Horze1)==0) & (isempty(Verte1)==0)
                plot(Horze1,Verte1, 'k o');
                if Horze1>xmax
                        xmax=Horze1;
                end;
                if Verte1>ymax
                        ymax=Verte1;
                end;
                hold on;
                set(gca,'Tag','Axes2');
        elseif twoearth==1 & (isempty(Horze1)==0) & (isempty(Verte1)==0) & (isempty(Horze2)==0) & (isempty(Verte2)==0)
                plot(Horze1,Verte1, 'k o');
                if Horze1>xmax
                        xmax=Horze1;
                end;
                if Verte1>ymax
                        ymax=Verte1;
                end;
                hold on;
                set(gca,'Tag','Axes2');
                plot(Horze2,Verte2, 'k o');
                if Horze2>xmax
                        xmax=Horze2;
                end;
                if Verte2>ymax
                        ymax=Verte2;
                end;
                hold on;
                set(gca,'Tag','Axes2');
        end;

        x=0;
        y=0:0.025:(ymax+1);
        plot(x,y, 'k');
        set(gca,'Tag','Axes2');

        set(gca,'Position',[842 61 170 310]);
        set(gca,'Color',[.6941 .6941 .6941]);
        axis([-(2*xmax+1) (2*xmax+1) 0 (1.25*ymax+1)]);
        ylabel('Vertical Spacing [m]', 'Fontsize',10);
        xlabel('Horizontal Spacing [m]', 'Fontsize',10);
        legend('phase a', 'phase b', 'phase c');
        title(Name, 'Fontsize', 10);

        hold off;

function PlotModGUITower(Horza,Horzb,Horzc,Horze1,Horze2,Verta,Vertb,Vertc,Verte1,Verte2)
        xmax=0;
        ymax=0;

        axes(findobj(findobj('Tag','Modify Tower'),'Tag','ModGUITowerAxes'))
        cla;
```

```
        plot(Horza,Verta, 'r o');
        if Horza>xmax
                xmax=Horza;
        end;
        if Vertb>ymax
                ymax=Verta;
        end;
        hold on;
        set(gca,'Tag','ModGUITowerAxes');

        plot(Horzb,Vertb,       'w o');
        if Horzb>xmax
                xmax=Horzb;
        end;
        if Vertb>ymax
                ymax=Vertb;
        end;
        hold on;
        set(gca,'Tag','ModGUITowerAxes');

        plot(Horzc,Vertc, 'b o');
        if Horzc>xmax
        xmax=Horzc;
        end;
        if Vertc>ymax
                ymax=Vertc;
        end;
        hold on;
        set(gca,'Tag','ModGUITowerAxes');

        plot(Horze1,Verte1, 'k o');
        if Horze1>xmax
        xmax=Horze1;
        end;
        if Verte1>ymax
                ymax=Verte1;
        end;
        hold on;
        set(gca,'Tag','ModGUITowerAxes');

        plot(Horze2,Verte2, 'k o');
        if Horze2>xmax
                xmax=Horze2;
        end;
        if Verte2>ymax
                ymax=Verte2;
        end;
        hold on;
        set(gca,'Tag','ModGUITowerAxes');

        x=0;
        y=0:0.025:(ymax+1);
        plot(x,y, 'k');
        set(gca,'Tag','ModGUITowerAxes');

        set(gca,'Position',[56 56 120 195]);
        set(gca,'Color',[.8 .8 .8]);
        axis([-(2*xmax+1) (2*xmax+1) 0 (1.25*ymax+1)]);
        ylabel('Vertical Spacing [m]', 'Fontsize',10);
        xlabel('Horizontal Spacing [m]', 'Fontsize',10);
        hold off;

function PlotAddGUITower(Horza,Horzb,Horzc,Horze1,Horze2,Verta,Vertb,Vertc,Verte1,Verte2)
        xmax=0;
        ymax=0;

        axes(findobj(findobj('Tag','Add Tower'),'Tag','AddGUITowerAxes'))
        cla;
        plot(Horza,Verta, 'r o');
        if Horza>xmax
                xmax=Horza;
        end;
        if Vertb>ymax
                ymax=Verta;
        end;
        hold on;
        set(gca,'Tag','AddGUITowerAxes');

        plot(Horzb,Vertb,       'w o');
        if Horzb>xmax
                xmax=Horzb;
        end;
        if Vertb>ymax
                ymax=Vertb;
        end;
        hold on;
        set(gca,'Tag','AddGUITowerAxes');

        plot(Horzc,Vertc, 'b o');
        if Horzc>xmax
                xmax=Horzc;
        end;
        if Vertc>ymax
                ymax=Vertc;
        end;
        hold on;
        set(gca,'Tag','AddGUITowerAxes');

        plot(Horze1,Verte1, 'k o');
        if Horze1>xmax
                xmax=Horze1;
```

```
        end;
        if Verte1>ymax
            ymax=Verte1;
        end;
        hold on;
        set(gca,'Tag','AddGUITowerAxes');

        plot(Horze2,Verte2, 'k o');
        if Horze2>xmax
            xmax=Horze2;
        end;
        if Verte2>ymax
            ymax=Verte2;
        end;
        hold on;
        set(gca,'Tag','AddGUITowerAxes');

        x=0;
        y=0:0.025:(ymax+1);
        plot(x,y, 'k');
        set(gca,'Tag','AddGUITowerAxes');

        set(gca,'Position',[56 56 120 195]);
        set(gca,'Color',[.8 .8 .8]);
        axis([-(2*xmax+1) (2*xmax+1) 0 (1.25*ymax+1)]);
        ylabel('Vertical Spacing [m]', 'Fontsize',10);
        xlabel('Horizontal Spacing [m]', 'Fontsize',10);
        hold off;

    function SetRB(Horze1,Verte1,Horze2,Verte2)
        if (isempty(Horze2)==1) & (isempty(Verte2)==1) & (isempty(Horze1)==1) & (isempty(Verte1)==1)
            set(findobj(findobj('Tag','OLPGUI'),'Tag','TwoEarthRB'),'Value',0);
            set(findobj(findobj('Tag','OLPGUI'),'Tag','TwoEarthRB'),'Enable','off');
            set(findobj(findobj('Tag','OLPGUI'),'Tag','OneEarthRB'),'Value',0);
            set(findobj(findobj('Tag','OLPGUI'),'Tag','OneEarthRB'),'Enable','off');
            set(findobj(findobj('Tag','OLPGUI'),'Tag','NoEarthRB'),'Value',1);
            set(findobj(findobj('Tag','OLPGUI'),'Tag','TwoEarthRB'),'Value',0);
            set(findobj(findobj('Tag','OLPGUI'),'Tag','Verte1'),'BackgroundColor',[.6941 .6941 .6941]);
            set(findobj(findobj('Tag','OLPGUI'),'Tag','Horze1'),'BackgroundColor',[.6941 .6941 .6941]);
            set(findobj(findobj('Tag','OLPGUI'),'Tag','Verte2'),'BackgroundColor',[.6941 .6941 .6941]);
            set(findobj(findobj('Tag','OLPGUI'),'Tag','Horze2'),'BackgroundColor',[.6941 .6941 .6941]);
            set(findobj(findobj('Tag','OLPGUI'),'Tag','Verte1'),'Enable','off');
            set(findobj(findobj('Tag','OLPGUI'),'Tag','Horze1'),'Enable','off');
            set(findobj(findobj('Tag','OLPGUI'),'Tag','Verte2'),'Enable','off');
            set(findobj(findobj('Tag','OLPGUI'),'Tag','Horze2'),'Enable','off');
            set(findobj(findobj('Tag','OLPGUI'),'Tag','Earth1mtext'),'Enable','off');
            set(findobj(findobj('Tag','OLPGUI'),'Tag','Earth2mtext'),'Enable','off');
            set(findobj(findobj('Tag','OLPGUI'),'Tag','Earth1text'),'Enable','off');
            set(findobj(findobj('Tag','OLPGUI'),'Tag','Earth2text'),'Enable','off');
        elseif (isempty(Horze2)==1) & (isempty(Verte2)==1) & (isempty(Horze1)==0) & (isempty(Verte1)==0)
            oneearth=get(findobj(findobj('Tag','OLPGUI'),'Tag','OneEarthRB'),'Value');
            noearth=get(findobj(findobj('Tag','OLPGUI'),'Tag','NoEarthRB'),'Value');
            if (oneearth==0) & (noearth==0)
                set(findobj(findobj('Tag','OLPGUI'),'Tag','NoEarthRB'),'Value',1);
                set(findobj(findobj('Tag','OLPGUI'),'Tag','Verte1'),'BackgroundColor',[.6941 .6941 .6941]);
                set(findobj(findobj('Tag','OLPGUI'),'Tag','Horze1'),'BackgroundColor',[.6941 .6941 .6941]);
                set(findobj(findobj('Tag','OLPGUI'),'Tag','Verte1'),'Enable','off');
                set(findobj(findobj('Tag','OLPGUI'),'Tag','Horze1'),'Enable','off');
                set(findobj(findobj('Tag','OLPGUI'),'Tag','Earth1mtext'),'Enable','off');
                set(findobj(findobj('Tag','OLPGUI'),'Tag','Earth1text'),'Enable','on');
            end;
            set(findobj(findobj('Tag','OLPGUI'),'Tag','TwoEarthRB'),'Value',0);
            set(findobj(findobj('Tag','OLPGUI'),'Tag','TwoEarthRB'),'Enable','off');
            set(findobj(findobj('Tag','OLPGUI'),'Tag','OneEarthRB'),'Enable','on');
            set(findobj(findobj('Tag','OLPGUI'),'Tag','Verte2'),'BackgroundColor',[.6941 .6941 .6941]);
            set(findobj(findobj('Tag','OLPGUI'),'Tag','Horze2'),'BackgroundColor',[.6941 .6941 .6941]);
            set(findobj(findobj('Tag','OLPGUI'),'Tag','Verte2'),'Enable','off');
            set(findobj(findobj('Tag','OLPGUI'),'Tag','Horze2'),'Enable','off');
            set(findobj(findobj('Tag','OLPGUI'),'Tag','Earth2mtext'),'Enable','off');
            set(findobj(findobj('Tag','OLPGUI'),'Tag','Earth2text'),'Enable','off');
        elseif (isempty(Horze2)==0) & (isempty(Verte2)==0) & (isempty(Horze1)==0) & (isempty(Verte1)==0)
            set(findobj(findobj('Tag','OLPGUI'),'Tag','TwoEarthRB'),'Enable','on');
            set(findobj(findobj('Tag','OLPGUI'),'Tag','OneEarthRB'),'Enable','on');
        end;
```

B.2 PSAT-RETICMASTER INTERFACE CODE

B.2.1 Interface.m

```
function Interface(action)

switch(action)

case 'Source'
        set(gcbo,'Value',1);
        set(findobj(findobj('Tag','PRIGUI'),'Tag','OtherRB'),'Value',0);
        set(findobj(findobj('Tag','PRIGUI'),'Tag','OtherListbox'),'Enable','off');
        set(findobj(findobj('Tag','PRIGUI'),'Tag','OtherListbox'),'BackgroundColor',[.6941 .6941 .6941]);
        set(findobj(findobj('Tag','PRIGUI'),'Tag','Sourcesr'),'Enable','on');
        set(findobj(findobj('Tag','PRIGUI'),'Tag','Sourcesr'),'BackgroundColor',[1 1 1]);
        set(findobj(findobj('Tag','PRIGUI'),'Tag','SourcesrVbase'),'Enable','on');
        set(findobj(findobj('Tag','PRIGUI'),'Tag','SourcesrVbase'),'BackgroundColor',[1 1 1]);
        set(findobj(findobj('Tag','PRIGUI'),'Tag','kVtext'),'Enable','on');

case 'Other'
        set(gcbo,'Value',1);
        set(findobj(findobj('Tag','PRIGUI'),'Tag','SourceRB'),'Value',0);
        set(findobj(findobj('Tag','PRIGUI'),'Tag','Sourcesr'),'Enable','off');
        set(findobj(findobj('Tag','PRIGUI'),'Tag','Sourcesr'),'BackgroundColor',[.6941 .6941 .6941]);
        set(findobj(findobj('Tag','PRIGUI'),'Tag','SourcesrVbase'),'Enable','off');
        set(findobj(findobj('Tag','PRIGUI'),'Tag','SourcesrVbase'),'BackgroundColor',[.6941 .6941 .6941]);
        set(findobj(findobj('Tag','PRIGUI'),'Tag','kVtext'),'Enable','off');
        set(findobj(findobj('Tag','PRIGUI'),'Tag','OtherListbox'),'Enable','on');
        set(findobj(findobj('Tag','PRIGUI'),'Tag','OtherListbox'),'BackgroundColor',[1 1 1]);

case 'LoadData'
        filename=get(findobj('Tag','filename'),'String');
        fid=fopen(filename, 'rt');
        data=fgetl(fid);
        data=fgetl(fid);
        i=1;

        while feof(fid) == 0
                data=fgetl(fid);
                name=fscanf(fid,'%s',1);
                Imagnitude=fscanf(fid,'%g',2);
                status=fseek(fid,1,'cof');
                Iangle=fscanf(fid,'%g',1);
                status=fseek(fid,1,'cof');
                Vmagnitude=fscanf(fid,'%g',1);
                status=fseek(fid,2,'cof');
                Vangle=fscanf(fid,'%g',1);
                status=fseek(fid,1,'cof');
                Source=fscanf(fid,'%s',1);
                status=fseek(fid,6,'cof');
                Vbase=fscanf(fid,'%g',1);
                Vunit=fscanf(fid,'%s',1);
                status=fseek(fid,1,'cof');
                technology=fscanf(fid,'%s',1);

                matchsource=strcmp(Source,'Source');
                if matchsource == 1
                        set(findobj('Tag','Sourcesr'),'String',name);
                        sourcesr(1)=Vmagnitude;
                        sourcesr(2)=Vangle;
                        sourcesr(3)=Imagnitude;
                        sourcesr(4)=Iangle;
                else
                retic(i).name=name;
                retic(i).Imagnitude=Imagnitude;
                retic(i).Iangle=Iangle;
                retic(i).Vmagnitude=Vmagnitude;
                retic(i).Vangle=Vangle;
                matchbase=strcmp(Vunit,'kV');
                matchtech=strcmp(technology,'Single');
                if (matchtech == 0) & (matchbase == 1)
                        Vbase=Vbase*1000;
                elseif (matchtech == 1) & (matchbase == 1)
                        Vbase=sqrt(3)*Vbase*1000;
                elseif (matchtech == 1) & (matchbase == 0)
                        Vbase=sqrt(3)*Vbase;
                end;
                retic(i).Vbase=Vbase;
                i=i+1;
                end;
        end;

        fclose(fid);

        for k=1:(length(retic)-1)
                NameSelect(k)={retic(k).name};
        end

        set(findobj('Tag','Sourcesr'),'UserData',sourcesr);
        set(findobj('Tag','LoadListbox'),'Value',1);
        set(findobj('Tag','OtherListbox'),'Value',1);
        set(findobj('Tag','LoadListbox'),'String',NameSelect);
        set(findobj('Tag','OtherListbox'),'String',NameSelect);
        set(findobj('Tag','PRIGUI'),'UserData',retic);
```

```
case 'Apply_Settings'
        SourceRB=get(findobj(findobj('Tag','PRIGUI'),'Tag','SourceRB'),'Value');
        OtherRB=get(findobj(findobj('Tag','PRIGUI'),'Tag','OtherRB'),'Value');

        SBASE=100e6;
        retic=get(findobj('Tag','PRIGUI'),'UserData');
        value=get(findobj('Tag','LoadListbox'),'Value');
        valuesr=get(findobj('Tag','OtherListbox'),'Value');

        %Calculate load end voltage (p.u.) (phase a)
        Vloadmagnitude=(retic(value).Vmagnitude)/100;
        Vloadangle=(retic(value).Vangle*pi)/180;
        Vload=Vloadmagnitude*cos(Vloadangle)+j*Vloadmagnitude*sin(Vloadangle);

        %Calculate load end current (p.u.) (phase a)
        Iloadmagnitude=(sqrt(3)*(retic(value).Imagnitude*retic(value).Vbase))/SBASE;
        Iloadangle=(retic(value).Iangle*pi)/180;
        Iload=Iloadmagnitude*cos(Iloadangle)+j*Iloadmagnitude*sin(Iloadangle);

        %Set outputs
        set(findobj('Tag','Vloadmagnitude'),'String',Vloadmagnitude);
        set(findobj('Tag','Vloadangle'),'String',sprintf('%3.2f',retic(value).Vangle));

        %--------------------------------------------------------------------

        if SourceRB==1
                sourcesr=get(findobj('Tag','Sourcesr'),'UserData');
                SourcesrVbase=str2num(get(findobj('Tag','SourcesrVbase'),'String'));

                %Calculate Sending-End or Receiving-End source voltage (p.u.) (phase a)
                Vsrmagnitude=sourcesr(1)/100;
                Vsrangle=(sourcesr(2)*pi)/180;
                Vsr=Vsrmagnitude*cos(Vsrangle)+j*Vsrmagnitude*sin(Vsrangle);

                %Calculate Sending-End or Receiving-End source current (p.u.) (phase a)
                Isrmagnitude=(sqrt(3)*sourcesr(3)*(SourcesrVbase*1000))/SBASE;
                Israngle=(sourcesr(4)*pi)/180;
                Isr=Isrmagnitude*cos(Israngle)+j*Isrmagnitude*sin(Israngle);

        elseif OtherRB==1
                %Calculate Sending-End or Receiving-End source voltage (p.u.) (phase a)
                Vsrmagnitude=(retic(valuesr).Vmagnitude)/100;
                Vsrangle=(retic(valuesr).Vangle*pi)/180;
                Vsr=Vsrmagnitude*cos(Vsrangle)+j*Vsrmagnitude*sin(Vsrangle);

                %Calculate Sending-End or Receiving-End source current (p.u.) (phase a)
                Isrmagnitude=(sqrt(3)*(retic(valuesr).Imagnitude*retic(valuesr).Vbase))/SBASE;
                Israngle=(retic(valuesr).Iangle*pi)/180;
                Isr=Isrmagnitude*cos(Israngle)+j*Isrmagnitude*sin(Israngle);
        end;

        %--------------------------------------------------------------------

        %Calculation of Zshunt, Zseries, Is or Ir, Zs or Zr and Vs or Vr

        Zshunt=Vload/(Isr-Iload);
        Zseries=(Vsr-Vload)/Isr;
        Zeq=(Zseries*Zshunt)/(Zseries+Zshunt);
        Veq=Zeq*(Vsr/Zseries);
        Ieq=(Veq-Vload)/Zeq;

        Veqmagnitude = abs(Veq);                        % magnitude
        Veqangle = (180*angle(Veq))/pi;                 % phase angle
        Ieqmagnitude = abs(Ieq);                        % magnitude
        Ieqangle = (180*angle(Ieq))/pi;                 % phase angle

        %Set outputs
        set(findobj('Tag','Veqmagnitude'),'String',sprintf('%3.4f',Veqmagnitude));
        set(findobj('Tag','Veqangle'),'String',sprintf('%3.2f',Veqangle));
        set(findobj('Tag','Ieqmagnitude'),'String',sprintf('%3.2e',Ieqmagnitude));
        set(findobj('Tag','Ieqangle'),'String',sprintf('%3.2f',Ieqangle));
        set(findobj('Tag','Zeq'),'String',num2str(Zeq,'%3.2e'));

case 'ClosePRI'
        fig = findobj('Tag','PRIGUI');
        saveas(figure(fig),get(fig,'FileName'),'fig');
        close(figure(fig));

end;
```

B.3 SUPPORT TECHNOLOGY AND DISPERSED GENERATION DATABASE CODE

B.3.1 Device.m

```
function Device(action)
%Main Device GUI Function

switch(action)

case 'DGAddGUI'
	set(gcbo,'Value',1);
	set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','ThyristorRB'),'Value',0);
	set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','MechanicalRB'),'Value',0);
	set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','PassiveRB'),'Value',0);
	set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','TransistorRB'),'Value',0);
case 'TransistorAddGUI'
	set(gcbo,'Value',1);
	set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','ThyristorRB'),'Value',0);
	set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','MechanicalRB'),'Value',0);
	set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','PassiveRB'),'Value',0);
	set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','DGRB'),'Value',0);
case 'ThyristorAddGUI'
	set(gcbo,'Value',1);
	set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','TransistorRB'),'Value',0);
	set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','MechanicalRB'),'Value',0);
	set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','PassiveRB'),'Value',0);
	set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','DGRB'),'Value',0);
case 'MechanicalAddGUI'
	set(gcbo,'Value',1);
	set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','ThyristorRB'),'Value',0);
	set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','TransistorRB'),'Value',0);
	set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','PassiveRB'),'Value',0);
	set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','DGRB'),'Value',0);
case 'PassiveAddGUI'
	set(gcbo,'Value',1);
	set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','ThyristorRB'),'Value',0);
	set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','MechanicalRB'),'Value',0);
	set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','TransistorRB'),'Value',0);
	set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','DGRB'),'Value',0);
case 'DGModGUI'
	set(gcbo,'Value',1);
	set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','TransistorRB'),'Value',0);
	set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ThyristorRB'),'Value',0);
	set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','MechanicalRB'),'Value',0);
	set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','PassiveRB'),'Value',0);
case 'TransistorModGUI'
	set(gcbo,'Value',1);
	set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DGRB'),'Value',0);
	set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ThyristorRB'),'Value',0);
	set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','MechanicalRB'),'Value',0);
	set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','PassiveRB'),'Value',0);
case 'ThyristorModGUI'
	set(gcbo,'Value',1);
	set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DGRB'),'Value',0);
	set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','TransistorRB'),'Value',0);
	set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','MechanicalRB'),'Value',0);
	set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','PassiveRB'),'Value',0);
case 'MechanicalModGUI'
	set(gcbo,'Value',1);
	set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DGRB'),'Value',0);
	set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ThyristorRB'),'Value',0);
	set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','TransistorRB'),'Value',0);
	set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','PassiveRB'),'Value',0);
case 'PassiveModGUI'
	set(gcbo,'Value',1);
	set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DGRB'),'Value',0);
	set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ThyristorRB'),'Value',0);
	set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','MechanicalRB'),'Value',0);
	set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','TransistorRB'),'Value',0);
case 'DGDeviceGUI'
	set(gcbo,'Value',1);
	set(findobj(findobj('Tag','DeviceGUI'),'Tag','TransistorRB'),'Value',0);
	set(findobj(findobj('Tag','DeviceGUI'),'Tag','ThyristorRB'),'Value',0);
	set(findobj(findobj('Tag','DeviceGUI'),'Tag','MechanicalRB'),'Value',0);
	set(findobj(findobj('Tag','DeviceGUI'),'Tag','PassiveRB'),'Value',0);
	set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceControl'),'BackgroundColor',[.6941 .6941 .6941]);
	set(findobj(findobj('Tag','DeviceGUI'),'Tag','DGDeviceControl'),'BackgroundColor',[1 1 1]);
case 'TransistorDeviceGUI'
	set(gcbo,'Value',1);
	set(findobj(findobj('Tag','DeviceGUI'),'Tag','DGRB'),'Value',0);
	set(findobj(findobj('Tag','DeviceGUI'),'Tag','ThyristorRB'),'Value',0);
	set(findobj(findobj('Tag','DeviceGUI'),'Tag','MechanicalRB'),'Value',0);
	set(findobj(findobj('Tag','DeviceGUI'),'Tag','PassiveRB'),'Value',0);
	set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceControl'),'BackgroundColor',[1 1 1]);
	set(findobj(findobj('Tag','DeviceGUI'),'Tag','DGDeviceControl'),'BackgroundColor',[.6941 .6941 .6941]);
case 'ThyristorDeviceGUI'
	set(gcbo,'Value',1);
	set(findobj(findobj('Tag','DeviceGUI'),'Tag','DGRB'),'Value',0);
	set(findobj(findobj('Tag','DeviceGUI'),'Tag','TransistorRB'),'Value',0);
	set(findobj(findobj('Tag','DeviceGUI'),'Tag','MechanicalRB'),'Value',0);
	set(findobj(findobj('Tag','DeviceGUI'),'Tag','PassiveRB'),'Value',0);
```

```
		set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceControl'),'BackgroundColor',[.6941 .6941 .6941]);
		set(findobj(findobj('Tag','DeviceGUI'),'Tag','DGDeviceControl'),'BackgroundColor',[.6941 .6941 .6941]);
	case 'MechanicalDeviceGUI'
		set(gcbo,'Value',1);
		set(findobj(findobj('Tag','DeviceGUI'),'Tag','DGRB'),'Value',0);
		set(findobj(findobj('Tag','DeviceGUI'),'Tag','ThyristorRB'),'Value',0);
		set(findobj(findobj('Tag','DeviceGUI'),'Tag','TransistorRB'),'Value',0);
		set(findobj(findobj('Tag','DeviceGUI'),'Tag','PassiveRB'),'Value',0);
		set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceControl'),'BackgroundColor',[.6941 .6941 .6941]);
		set(findobj(findobj('Tag','DeviceGUI'),'Tag','DGDeviceControl'),'BackgroundColor',[.6941 .6941 .6941]);
	case 'PassiveDeviceGUI'
		set(gcbo,'Value',1);
		set(findobj(findobj('Tag','DeviceGUI'),'Tag','DGRB'),'Value',0);
		set(findobj(findobj('Tag','DeviceGUI'),'Tag','ThyristorRB'),'Value',0);
		set(findobj(findobj('Tag','DeviceGUI'),'Tag','MechanicalRB'),'Value',0);
		set(findobj(findobj('Tag','DeviceGUI'),'Tag','TransistorRB'),'Value',0);
		set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceControl'),'BackgroundColor',[.6941 .6941 .6941]);
		set(findobj(findobj('Tag','DeviceGUI'),'Tag','DGDeviceControl'),'BackgroundColor',[.6941 .6941 .6941]);
	case 'SinglephaseAddGUI'
		set(gcbo,'Value',1);
		set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','3phaseRB'),'Value',0);
	case 'ThreephaseAddGUI'
		set(gcbo,'Value',1);
		set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','1phaseRB'),'Value',0);
	case 'SinglephaseModGUI'
		set(gcbo,'Value',1);
		set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','3phaseRB'),'Value',0);
	case 'ThreephaseModGUI'
		set(gcbo,'Value',1);
		set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','1phaseRB'),'Value',0);
	case 'SinglephaseDeviceGUI'
		set(gcbo,'Value',1);
		set(findobj(findobj('Tag','DeviceGUI'),'Tag','3phaseRB'),'Value',0);
	case 'ThreephaseDeviceGUI'
		set(gcbo,'Value',1);
		set(findobj(findobj('Tag','DeviceGUI'),'Tag','1phaseRB'),'Value',0);

	case 'Calculate'
		Sbase=str2num(get(findobj(findobj('Tag','DeviceGUI'),'Tag','Sbase'),'String'));
		Vbase=str2num(get(findobj(findobj('Tag','DeviceGUI'),'Tag','Vbase'),'String'));
		Ppu=str2num(get(findobj(findobj('Tag','DeviceGUI'),'Tag','Ppu'),'String'));
		Qpu=str2num(get(findobj(findobj('Tag','DeviceGUI'),'Tag','Qpu'),'String'));

		set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'Value',1);
		set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String','');

		if isempty(Ppu) & ~isempty(Qpu)
			Q=Qpu*Sbase;
			S=Q;
			set(findobj(findobj('Tag','DeviceGUI'),'Tag','S'),'String',num2str(S));
			set(findobj(findobj('Tag','DeviceGUI'),'Tag','Q'),'String',num2str(Q));
			set(findobj(findobj('Tag','DeviceGUI'),'Tag','P'),'String','');
		elseif ~isempty(Ppu) & ~isempty(Qpu)
			P=Ppu*Sbase;
			Q=Qpu*Sbase;
			S=abs(P+j*Q);
			set(findobj(findobj('Tag','DeviceGUI'),'Tag','S'),'String',num2str(S));
			set(findobj(findobj('Tag','DeviceGUI'),'Tag','P'),'String',num2str(P));
			set(findobj(findobj('Tag','DeviceGUI'),'Tag','Q'),'String',num2str(Q));
		elseif ~isempty(Ppu) & isempty(Qpu)
			S=Ppu*Sbase;
			P=S;
			set(findobj(findobj('Tag','DeviceGUI'),'Tag','S'),'String',num2str(S));
			set(findobj(findobj('Tag','DeviceGUI'),'Tag','P'),'String',num2str(P));
			set(findobj(findobj('Tag','DeviceGUI'),'Tag','Q'),'String','');
		else isempty(Ppu) & isempty(Qpu)
			set(findobj(findobj('Tag','DeviceGUI'),'Tag','S'),'String','');
			set(findobj(findobj('Tag','DeviceGUI'),'Tag','P'),'String','');
			set(findobj(findobj('Tag','DeviceGUI'),'Tag','Q'),'String','');
		end;

		set(findobj(findobj('Tag','DeviceGUI'),'Tag','V'),'String',num2str(Vbase));
		set(findobj(findobj('Tag','DeviceGUI'),'Tag','Info'),'String','fit device');

	case 'Display_Device'
		%Display Selected Conductor Parameters
		DData=get(findobj(findobj('Tag','DeviceGUI'),'Tag','DGRB'),'UserData');

		value=get(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'Value');

		set(findobj(findobj('Tag','DeviceGUI'),'Tag','Info'),'String',DData(value).Info);

		set(findobj(findobj('Tag','DeviceGUI'),'Tag','V'),'String',DData(value).Voltage);
		set(findobj(findobj('Tag','DeviceGUI'),'Tag','P'),'String',DData(value).P);
		set(findobj(findobj('Tag','DeviceGUI'),'Tag','Q'),'String',DData(value).Q);
		P=str2num(get(findobj(findobj('Tag','DeviceGUI'),'Tag','P'),'String'));
		Q=str2num(get(findobj(findobj('Tag','DeviceGUI'),'Tag','Q'),'String'));
		S=abs(P+j*Q);
		set(findobj(findobj('Tag','DeviceGUI'),'Tag','S'),'String',num2str(S));

	case 'Fit_Device'
		ShuntStructure=get(findobj(findobj('Tag','DeviceGUI'),'Tag','MechanicalRB'),'UserData');
		SeriesStructure=get(findobj(findobj('Tag','DeviceGUI'),'Tag','PassiveRB'),'UserData');
		SSStructure=get(findobj(findobj('Tag','DeviceGUI'),'Tag','ThyristorRB'),'UserData');
		ILStructure=get(findobj(findobj('Tag','DeviceGUI'),'Tag','TransistorRB'),'UserData');

		DeviceConfig=get(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceConfig'),'String');
		V=str2num(get(findobj(findobj('Tag','DeviceGUI'),'Tag','V'),'String'));
		P=str2num(get(findobj(findobj('Tag','DeviceGUI'),'Tag','P'),'String'));
		Q=str2num(get(findobj(findobj('Tag','DeviceGUI'),'Tag','Q'),'String'));
		EF=get(findobj(findobj('Tag','DeviceGUI'),'Tag','EnergystorageCB'),'Value');
```

```
    SF=get(findobj(findobj('Tag','DeviceGUI'),'Tag','1phaseRB'),'Value');
    TF=get(findobj(findobj('Tag','DeviceGUI'),'Tag','3phaseRB'),'Value');
    Passive=get(findobj(findobj('Tag','DeviceGUI'),'Tag','PassiveRB'),'Value');
    Mechanical=get(findobj(findobj('Tag','DeviceGUI'),'Tag','MechanicalRB'),'Value');
    Thyristor=get(findobj(findobj('Tag','DeviceGUI'),'Tag','ThyristorRB'),'Value');
    Transistor=get(findobj(findobj('Tag','DeviceGUI'),'Tag','TransistorRB'),'Value');
    DG=get(findobj(findobj('Tag','DeviceGUI'),'Tag','DGRB'),'Value');

    %Initialize Variables
    k=1;
    set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'Value',1);
    Devices={};
    set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);

    %Find Devices
    %Shunt Configuration
    if (strcmp(DeviceConfig,'Shunt')==1);
        for i=1:length(ShuntStructure)
            if isempty(V)|V>str2num(ShuntStructure(i).Voltage)
                VF=0;
            else
                VF=1;
            end;
            if isempty(P)|P>str2num(ShuntStructure(i).P)
                PF=0;
            else
                PF=1;
            end;
            if isempty(Q)|Q>str2num(ShuntStructure(i).Q)
                QF=0;
            else
                QF=1;
            end;
            A=[VF PF QF EF SF TF];
            B=[ShuntStructure(i).VF ShuntStructure(i).PF ShuntStructure(i).QF ShuntStructure(i)
                            Energy ShuntStructure(i).Singlephase ShuntStructure(i).Threephase];
            if (ShuntStructure(i).DeviceTech==1) & (A==B) & (Passive==1);              Shunt-Passive
                DData(k)=ShuntStructure(i);
                for l=1:length(DData)
                    Devices(l)={DData(l).Name};
                    set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
                end
            k=k+1;
            elseif (ShuntStructure(i).DeviceTech==2) & (A==B) & (Mechanical==1);        %Shunt-Mechanical
                DData(k)=ShuntStructure(i);
                for l=1:length(DData)
                    Devices(l)={DData(l).Name};
                    set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
                end
                k=k+1;
            elseif (ShuntStructure(i).DeviceTech==3) & (A==B) & (Thyristor==1);         %Shunt-Thyristor
                DData(k)=ShuntStructure(i);
                for l=1:length(DData)
                    Devices(l)={DData(l).Name};
                    set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
                end
                k=k+1;
            elseif (ShuntStructure(i).DeviceTech==4) & (A==B) & (Transistor==1);        %Shunt-In-line
                DData(k)=ShuntStructure(i);
                for l=1:length(DData)
                    Devices(l)={DData(l).Name};
                    set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
                end
                k=k+1;
            elseif (ShuntStructure(i).DeviceTech==5) & (A==B) & (DG==1);                %Distributed Generation
                DData(k)=ShuntStructure(i);
                for l=1:length(DData)
                    Devices(l)={DData(l).Name};
                    set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
                end
                k=k+1;
            elseif isempty(Devices)
                Devices={};
                set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
            end;
        end;

    %Series Configuration
    elseif (strcmp(DeviceConfig,'Series')==1);
        for i=1:length(SeriesStructure)
            if isempty(V)|V>str2num(SeriesStructure(i).Voltage)
                VF=0;
            else
                VF=1;
            end;
            if isempty(P)|P>str2num(SeriesStructure(i).P)
                PF=0;
            else
                PF=1;
            end;
            if isempty(Q)|Q>str2num(SeriesStructure(i).Q)
                QF=0;
            else
                QF=1;
            end;
            A=[VF PF QF EF SF TF];
            B=[SeriesStructure(i).VF SeriesStructure(i).PF SeriesStructure(i).QF SeriesStructure(i)
                            Energy SeriesStructure(i).Singlephase SeriesStructure(i).Threephase];
            if (SeriesStructure(i).DeviceTech==1) & (A==B) & (Passive==1);             %Series-Passive
                DData(k)=SeriesStructure(i);
                for l=1:length(DData)
```

```
                    Devices(l)={DData(l).Name};
                    set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
                end
            k=k+1;
            elseif (SeriesStructure(i).DeviceTech==2) & (A==B) & (Mechanical==1);          %Series-Mechanical
                DData(k)=SeriesStructure(i);
                for l=1:length(DData)
                    Devices(l)={DData(l).Name};
                    set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
                end
                k=k+1;
            elseif (SeriesStructure(i).DeviceTech==3) & (A==B) & (Thyristor==1);           %Series-Thyristor
                DData(k)=SeriesStructure(i);
                for l=1:length(DData)
                    Devices(l)={DData(l).Name};
                    set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
                end
                k=k+1;
            elseif (SeriesStructure(i).DeviceTech==4) & (A==B) & (Transistor==1);          %Series-Transistor
                DData(k)=SeriesStructure(i);
                for l=1:length(DData)
                    Devices(l)={DData(l).Name};
                    set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
                end
                k=k+1;
            elseif isempty(Devices)
                Devices={};
                set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
            end;
        end;

    %Series-Shunt Configuration
    elseif (strcmp(DeviceConfig,'Series-shunt')==1);
        for i=1:length(SSStructure)
            if isempty(V)|V>str2num(SSStructure(i).Voltage)
                VF=0;
            else
                VF=1;
            end;
            if isempty(P)|P>str2num(SSStructure(i).P)
                PF=0;
            else
                PF=1;
            end;
            if isempty(Q)|Q>str2num(SSStructure(i).Q)
                QF=0;
            else
                QF=1;
            end;
            A=[VF PF QF EF SF TF];
            B=[SSStructure(i).VF SSStructure(i).PF SSStructure(i).QF SSStructure(i).Energy
                        SSStructure(i).Singlephase SSStructure(i).Threephase];
            if (SSStructure(i).DeviceTech==1) & (A==B) & (Passive==1);                      %Series-shunt-Passive
                DData(k)=SSStructure(i);
                for l=1:length(DData)
                    Devices(l)={DData(l).Name};
                    set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
                end
                k=k+1;
            elseif (SSStructure(i).DeviceTech==2) & (A==B) & (Mechanical==1);               %Series-shunt-Mechanical
                DData(k)=SSStructure(i);
                for l=1:length(DData)
                    Devices(l)={DData(l).Name};
                    set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
                end
                k=k+1;
            elseif (SSStructure(i).DeviceTech==3) & (A==B) & (Thyristor==1);                %Series-shunt-Thyristor
                DData(k)=SSStructure(i);
                for l=1:length(DData)
                    Devices(l)={DData(l).Name};
                    set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
                end
                k=k+1;
            elseif (SSStructure(i).DeviceTech==4) & (A==B) & (Transistor==1);               %Series-shunt-Transistor
                DData(k)=SSStructure(i);
                for l=1:length(DData)
                    Devices(l)={DData(l).Name};
                    set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
                end
                k=k+1;
            elseif isempty(Devices)
                Devices={};
                set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
            end;
        end;

    %In-line Configuration
    elseif (strcmp(DeviceConfig,'In-line')==1);
        for i=1:length(ILStructure)
            if isempty(V)|V>str2num(ILStructure(i).Voltage)
                VF=0;
            else
                VF=1;
            end;
            if isempty(P)|P>str2num(ILStructure(i).P)
                PF=0;
            else
                PF=1;
            end;
            if isempty(Q)|Q>str2num(ILStructure(i).Q)
                QF=0;
```

```
                else
                    QF=1;
                end;
                A=[VF PF QF EF SF TF];
                B=[ILStructure(i).VF ILStructure(i).PF ILStructure(i).QF ILStructure(i)
                                        Energy ILStructure(i).Singlephase ILStructure(i).Threephase];
                if (ILStructure(i).DeviceTech==1) & (A==B) & (Passive==1);                    %In-line-Passive
                    DData(k)=ILStructure(i);
                    for l=1:length(DData)
                        Devices(l)={DData(l).Name};
                        set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
                    end
                    k=k+1;
                elseif (ILStructure(i).DeviceTech==2) & (A==B) & (Mechanical==1);             %In-line-Mechanical
                    DData(k)=ILStructure(i);
                    for l=1:length(DData)
                        Devices(l)={DData(l).Name};
                        set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
                    end
                    k=k+1;
                elseif (ILStructure(i).DeviceTech==3) & (A==B) & (Thyristor==1);              %In-line-Thyristor
                    DData(k)=ILStructure(i);
                    for l=1:length(DData)
                        Devices(l)={DData(l).Name};
                        set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
                    end
                    k=k+1;
                elseif (ILStructure(i).DeviceTech==4) & (A==B) & (Transistor==1);             %In-line-Transistor
                    DData(k)=ILStructure(i);
                    for l=1:length(DData)
                        Devices(l)={DData(l).Name};
                        set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
                    end
                    k=k+1;
                elseif isempty(Devices)
                    Devices={};
                    set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
                end;
            end;
        else
            Devices={};
        end;

        %SetDeviceNames(Devices)
        value=get(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'Value');

        if isempty(Devices)
            set(findobj(findobj('Tag','DeviceGUI'),'Tag','V'),'String',num2str(V));
            set(findobj(findobj('Tag','DeviceGUI'),'Tag','P'),'String',num2str(P));
            set(findobj(findobj('Tag','DeviceGUI'),'Tag','Q'),'String',num2str(Q));
            S=abs(P+j*Q);
            set(findobj(findobj('Tag','DeviceGUI'),'Tag','S'),'String',num2str(S));
            set(findobj(findobj('Tag','DeviceGUI'),'Tag','Info'),'String','no device');
        else
            set(findobj(findobj('Tag','DeviceGUI'),'Tag','DGRB'),'UserData',DData);
            set(findobj(findobj('Tag','DeviceGUI'),'Tag','Info'),'String',DData(value).Info);
            set(findobj(findobj('Tag','DeviceGUI'),'Tag','V'),'String',DData(value).Voltage);
            set(findobj(findobj('Tag','DeviceGUI'),'Tag','P'),'String',DData(value).P);
            set(findobj(findobj('Tag','DeviceGUI'),'Tag','Q'),'String',DData(value).Q);
            P=str2num(get(findobj(findobj('Tag','DeviceGUI'),'Tag','P'),'String'));
            Q=str2num(get(findobj(findobj('Tag','DeviceGUI'),'Tag','Q'),'String'));
            S=abs(P+j*Q);
            set(findobj(findobj('Tag','DeviceGUI'),'Tag','S'),'String',num2str(S));
        end;

    case 'Add_Device'
        SeriesStructure=get(findobj(findobj('Tag','DeviceGUI'),'Tag','PassiveRB'),'UserData');
        ShuntStructure=get(findobj(findobj('Tag','DeviceGUI'),'Tag','MechanicalRB'),'UserData');
        SSStructure=get(findobj(findobj('Tag','DeviceGUI'),'Tag','ThyristorRB'),'UserData');
        ILStructure=get(findobj(findobj('Tag','DeviceGUI'),'Tag','TransistorRB'),'UserData');

        AddName=get(findobj(findobj('Tag','AddDeviceGUI'),'Tag','AddName'),'String');
        AddP=get(findobj(findobj('Tag','AddDeviceGUI'),'Tag','AddDeviceP'),'String');
        AddQ=get(findobj(findobj('Tag','AddDeviceGUI'),'Tag','AddDeviceQ'),'String');
        AddV=get(findobj(findobj('Tag','AddDeviceGUI'),'Tag','AddDeviceV'),'String');
        Energy=get(findobj(findobj('Tag','AddDeviceGUI'),'Tag','EnergystorageCB'),'Value');
        Singlephase=get(findobj(findobj('Tag','AddDeviceGUI'),'Tag','1phaseRB'),'Value');
        Threephase=get(findobj(findobj('Tag','AddDeviceGUI'),'Tag','3phaseRB'),'Value');
        DeviceConfig=get(findobj(findobj('Tag','AddDeviceGUI'),'Tag','DeviceConfigMenu'),'Value');
        DeviceInfo=get(findobj(findobj('Tag','AddDeviceGUI'),'Tag','InfoEdit'),'String');

        Transistor=get(findobj(findobj('Tag','AddDeviceGUI'),'Tag','TransistorRB'),'Value');
        Thyristor=get(findobj(findobj('Tag','AddDeviceGUI'),'Tag','ThyristorRB'),'Value');
        Mechanical=get(findobj(findobj('Tag','AddDeviceGUI'),'Tag','MechanicalRB'),'Value');
        Passive=get(findobj(findobj('Tag','AddDeviceGUI'),'Tag','PassiveRB'),'Value');
        DG=get(findobj(findobj('Tag','AddDeviceGUI'),'Tag','DGRB'),'Value');

        if Passive==1
            DeviceTech=1;
        elseif Mechanical==1
            DeviceTech=2;
        elseif Thyristor==1
            DeviceTech=3;
        elseif Transistor==1
            DeviceTech=4;
        else
            DeviceTech=5;
        end;

        if isempty(AddV)
            VF=0;
```

```
		else
			VF=1;
		end;

		if (isempty(AddP)==1) & (isempty(AddQ)==0)
			AddS=j*str2num(AddQ);
			PF=0;
			QF=1;
		elseif (isempty(AddP)==0) & (isempty(AddQ)==0)
			AddS=str2num(AddP)+j*str2num(AddQ);
			PF=1;
			QF=1;
		elseif (isempty(AddP)==0) & (isempty(AddQ)==1)
			AddS=str2num(AddP);
			PF=1;
			QF=0;
		else (isempty(AddP)==1) & (isempty(AddQ)==1)
			PF=0;
			QF=0;
		end;

		if (PF==0) & (QF==0)
			set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','AddDeviceS'),'String','');
		else
			set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','AddDeviceS'),'String',num2str(AddS));
		end;

		if (DeviceConfig==1)
			sz=size(SeriesStructure);
			SeriesStructure(sz(2)+1)=struct('DeviceTech',DeviceTech,'Name',AddName,'Voltage',AddV,'VF',VF,'P',AddP,
		'PF',PF,'Q',AddQ,'QF',QF,'Energy',Energy,'Singlephase',Singlephase,'Threephase',Threephase,'Info',DeviceInfo);
			set(findobj(findobj('Tag','DeviceGUI'),'Tag','PassiveRB'),'UserData',SeriesStructure);
		elseif (DeviceConfig==2)
			sz=size(ShuntStructure);
			ShuntStructure(sz(2)+1)=struct('DeviceTech',DeviceTech,'Name',AddName,'Voltage',AddV,'VF',VF,'P',AddP,
		'PF',PF,'Q',AddQ,'QF',QF,'Energy',Energy,'Singlephase',Singlephase,'Threephase',Threephase,'Info',DeviceInfo);
			set(findobj(findobj('Tag','DeviceGUI'),'Tag','MechanicalRB'),'UserData',ShuntStructure);
		elseif (DeviceConfig==3)
			sz=size(SSStructure);
			SSStructure(sz(2)+1)=struct('DeviceTech',DeviceTech,'Name',AddName,'Voltage',AddV,'VF',VF,'P',AddP,'PF',
			PF,'Q',AddQ,'QF',QF,'Energy',Energy,'Singlephase',Singlephase,'Threephase',Threephase,'Info',DeviceInfo);
			set(findobj(findobj('Tag','DeviceGUI'),'Tag','ThyristorRB'),'UserData',SSStructure);
		elseif (DeviceConfig==4)
			sz=size(ILStructure);
			ILStructure(sz(2)+1)=struct('DeviceTech',DeviceTech,'Name',AddName,'Voltage',AddV,'VF',VF,'P',AddP,'PF',
			PF,'Q',AddQ,'QF',QF,'Energy',Energy,'Singlephase',Singlephase,'Threephase',Threephase,'Info',DeviceInfo);
			set(findobj(findobj('Tag','DeviceGUI'),'Tag','TransistorRB'),'UserData',ILStructure);
		end;

		fig = findobj('Tag','DeviceGUI');
		saveas(figure(fig),get(fig,'FileName'),'fig');

	case 'GetStructure'
		set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'Value',1);
		DeviceConfig=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceConfigMenu'),'Value');
		if DeviceConfig==1
			SeriesStructure=get(findobj(findobj('Tag','DeviceGUI'),'Tag','PassiveRB'),'UserData');
			for l=1:length(SeriesStructure)
				Name(l)={SeriesStructure(l).Name};
			end
			set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'String',Name);
			set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'UserData',SeriesStructure);
		elseif DeviceConfig==2
			ShuntStructure=get(findobj(findobj('Tag','DeviceGUI'),'Tag','MechanicalRB'),'UserData');
			for l=1:length(ShuntStructure)
				Name(l)={ShuntStructure(l).Name};
			end
			set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'String',Name);
			set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'UserData',ShuntStructure);
		elseif DeviceConfig==3
			SSStructure=get(findobj(findobj('Tag','DeviceGUI'),'Tag','ThyristorRB'),'UserData');
			for l=1:length(SSStructure)
				Name(l)={SSStructure(l).Name};
			end
			set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'String',Name);
			set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'UserData',SSStructure);
		elseif DeviceConfig==4
			ILStructure=get(findobj(findobj('Tag','DeviceGUI'),'Tag','TransistorRB'),'UserData');
			for l=1:length(ILStructure)
				Name(l)={ILStructure(l).Name};
			end
			set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'String',Name);
			set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'UserData',ILStructure);
			end;

	case 'DisplayDevice'
		value=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'Value');
		devicestruct=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'UserData');
		VF=devicestruct(value).VF;
		DeviceTech=devicestruct(value).DeviceTech;
		if DeviceTech==1
			set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','PassiveRB'),'Value',1);
			set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','MechanicalRB'),'Value',0);
			set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ThyristorRB'),'Value',0);
			set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','TransistorRB'),'Value',0);
			set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DGRB'),'Value',0)
		elseif DeviceTech==2
			set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','MechanicalRB'),'Value',1);
			set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','PassiveRB'),'Value',0);
			set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ThyristorRB'),'Value',0);
			set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','TransistorRB'),'Value',0);
```

```
                set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DGRB'),'Value',0)
            elseif DeviceTech==3
                set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ThyristorRB'),'Value',1);
                set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','MechanicalRB'),'Value',0);
                set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','PassiveRB'),'Value',0);
                set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','TransistorRB'),'Value',0);
                set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DGRB'),'Value',0)
            elseif DeviceTech==4
                set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','TransistorRB'),'Value',1);
                set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','MechanicalRB'),'Value',0);
                set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ThyristorRB'),'Value',0);
                set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','PassiveRB'),'Value',0);
                set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DGRB'),'Value',0)
            else
                set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DGRB'),'Value',1);
                set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','TransistorRB'),'Value',0);
                set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','MechanicalRB'),'Value',0);
                set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ThyristorRB'),'Value',0);
                set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','PassiveRB'),'Value',0);
            end;

            if VF==1
                set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceV'),'String',devicestruct(value).Voltage);
            else
                set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceV'),'String','');
            end;
            PF=devicestruct(value).PF;
            QF=devicestruct(value).QF;

            if (PF==1) & (QF==1)
                set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceP'),'String',devicestruct(value).P);
                set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceQ'),'String',devicestruct(value).Q);
                S=str2num(devicestruct(value).P)+j*str2num(devicestruct(value).Q);
                set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceS'),'String',num2str(S));
            elseif (PF==1) & (QF==0)
                set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceP'),'String',devicestruct(value).P);
                set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceQ'),'String','');
                set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceS'),'String',devicestruct(value).P);
            elseif (PF==0) & (QF==1)
                set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceP'),'String','');
                set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceQ'),'String',devicestruct(value).Q);
                S=j*str2num(devicestruct(value).Q);
                set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceS'),'String',num2str(S));
            else
                set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceP'),'String','');
                set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceQ'),'String','');
                set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceS'),'String','');
            end;

            set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModName'),'String',devicestruct(value).Name);
            set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModInfo'),'String',devicestruct(value).Info);
            set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','EnergystorageCB'),'Value',devicestruct(value).Energy);
            set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','1phaseRB'),'Value',devicestruct(value).Singlephase);
            set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','3phaseRB'),'Value',devicestruct(value).Threephase);

        case 'Modify'
            value=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'Value');
            devicestruct=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'UserData');

            ModName=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModName'),'String');
            ModInfo=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModInfo'),'String');
            ModP=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceP'),'String');
            ModQ=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceQ'),'String');
            ModV=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceV'),'String');

            Energy=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','EnergystorageCB'),'Value');
            Singlephase=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','1phaseRB'),'Value');
            Threephase=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','3phaseRB'),'Value');

            Transistor=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','TransistorRB'),'Value');
            Thyristor=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ThyristorRB'),'Value');
            Mechanical=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','MechanicalRB'),'Value');
            Passive=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','PassiveRB'),'Value');
            DG=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DGRB'),'Value');

            if Passive==1
                DeviceTech=1;
            elseif Mechanical==1
                DeviceTech=2;
            elseif Thyristor==1
                DeviceTech=3;
            elseif Transistor==1
                DeviceTech=4;
            else
                DeviceTech=5;
            end;

            if isempty(ModV)
                VF=0;
            else
                VF=1;
            end;

            if (isempty(ModP)==1) & (isempty(ModQ)==0)
                ModS=j*str2num(ModQ);
                PF=0;
                QF=1;
            elseif (isempty(ModP)==0) & (isempty(ModQ)==0)
                ModS=str2num(ModP)+j*str2num(ModQ);
                PF=1;
                QF=1;
```

```
        elseif (isempty(ModP)==0) & (isempty(ModQ)==1)
            ModS=str2num(ModP);
            PF=1;
            QF=0;
        else (isempty(ModP)==1) & (isempty(ModQ)==1)
            PF=0;
            QF=0;
        end

        if (PF==0) & (QF==0)
            set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceS'),'String','');
        else
            set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceS'),'String',num2str(ModS));
        end;

        DeviceConfig=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceConfigMenu'),'Value');
        SeriesStructure=get(findobj(findobj('Tag','DeviceGUI'),'Tag','PassiveRB'),'UserData');
        ShuntStructure=get(findobj(findobj('Tag','DeviceGUI'),'Tag','MechanicalRB'),'UserData');
        SSStructure=get(findobj(findobj('Tag','DeviceGUI'),'Tag','ThyristorRB'),'UserData');
        ILStructure=get(findobj(findobj('Tag','DeviceGUI'),'Tag','TransistorRB'),'UserData');
        if (DeviceConfig==1)
            SeriesStructure(value)=struct('DeviceTech',DeviceTech,'Name',ModName,'Voltage',ModV,'VF',VF,'P',ModP,
        'PF',PF,'Q',ModQ,'QF',QF,'Energy',Energy,'Singlephase',Singlephase,'Threephase',Threephase,'Info',ModInfo);
            for l=1:length(SeriesStructure)
                Name(l)={SeriesStructure(l).Name};
            end
            set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'String',Name);
            set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'UserData',SeriesStructure);
            set(findobj(findobj('Tag','DeviceGUI'),'Tag','PassiveRB'),'UserData',SeriesStructure);
        elseif (DeviceConfig==2)
            ShuntStructure(value)=struct('DeviceTech',DeviceTech,'Name',ModName,'Voltage',ModV,'VF',VF,'P',ModP,'PF'
            ,PF,'Q',ModQ,'QF',QF,'Energy',Energy,'Singlephase',Singlephase,'Threephase',Threephase,'Info',ModInfo);
            for l=1:length(ShuntStructure)
                Name(l)={ShuntStructure(l).Name};
            end
            set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'String',Name);
            set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'UserData',ShuntStructure);
            set(findobj(findobj('Tag','DeviceGUI'),'Tag','MechanicalRB'),'UserData',ShuntStructure);
        elseif (DeviceConfig==3)
            SSStructure(value)=struct('DeviceTech',DeviceTech,'Name',ModName,'Voltage',ModV,'VF',VF,'P',ModP,'PF',
            PF,'Q',ModQ,'QF',QF,'Energy',Energy,'Singlephase',Singlephase,'Threephase',Threephase,'Info',ModInfo);
            for l=1:length(SSStructure)
                Name(l)={SSStructure(l).Name
            end;
            set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'String',Name);
            set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'UserData',SSStructure);
            set(findobj(findobj('Tag','DeviceGUI'),'Tag','ThyristorRB'),'UserData',SSStructure);
        elseif (DeviceConfig==4)
            ILStructure(value)=struct('DeviceTech',DeviceTech,'Name',ModName,'Voltage',ModV,'VF',VF,'P',ModP,'PF',
            PF,'Q',ModQ,'QF',QF,'Energy',Energy,'Singlephase',Singlephase,'Threephase',Threephase,'Info',ModInfo);
            for l=1:length(ILStructure)
                Name(l)={ILStructure(l).Name};
            end
            set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'String',Name);
            set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'UserData',ILStructure);
            set(findobj(findobj('Tag','DeviceGUI'),'Tag','TransistorRB'),'UserData',ILStructure);
        end;
        fig = findobj('Tag','DeviceGUI');
        saveas(figure(fig),get(fig,'FileName'),'fig');

    case 'OpenAddDevice'
        fig=findobj('Tag','AddDeviceGUI');
        if isempty(fig)==1
            open('AddDevice.fig');
        end;
    case 'OpenModifyDevice'
        fig=findobj('Tag','ModDeviceGUI');
        if isempty(fig)==1
            open('ModifyDevice.fig');
        end;

    case 'CloseAddDevice'
        fig = findobj('Tag','AddDeviceGUI');
        close(figure(fig));

    case 'CloseModDevice'
        fig = findobj('Tag','ModDeviceGUI');
        close(figure(fig));

    case 'CloseDeviceLib'
        fig = findobj('Tag','DeviceGUI');
        saveas(figure(fig),get(fig,'FileName'),'fig');
        close(figure(fig));

    end;
```

B.4 TESAT CODE

B.4.1 LFAC.m

```
function lfac(action)
%Main Loadflow for a single three-phase ac line GUI

switch(action)

case 'VoltageRegulation'
	set(gcbo,'Value',1);
	set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcontunit'),'String','%');
	set(findobj(findobj('Tag','lfacGUI'),'Tag','lossrb'),'Value',0);
	set(findobj(findobj('Tag','lfacGUI'),'Tag','cclrb'),'Value',0);
	set(findobj(findobj('Tag','lfacGUI'),'Tag','ATIrb'),'Value',0);
	set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value',0);
	set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcont'),'String','0');
case 'Loss'
	set(gcbo,'Value',1);
	set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcontunit'),'String','kW');
	set(findobj(findobj('Tag','lfacGUI'),'Tag','vrrb'),'Value',0);
	set(findobj(findobj('Tag','lfacGUI'),'Tag','cclrb'),'Value',0);
	set(findobj(findobj('Tag','lfacGUI'),'Tag','ATIrb'),'Value',0);
	set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value',10);
	set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcont'),'String','10');

case 'CCL'
	set(gcbo,'Value',1);
	set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcontunit'),'String','R');
	set(findobj(findobj('Tag','lfacGUI'),'Tag','vrrb'),'Value',0);
	set(findobj(findobj('Tag','lfacGUI'),'Tag','lossrb'),'Value',0);
	set(findobj(findobj('Tag','lfacGUI'),'Tag','ATIrb'),'Value',0);
	set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value',10);
	set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcont'),'String','100000');

case 'ATI'
	set(gcbo,'Value',1);
	set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcontunit'),'String','');
	set(findobj(findobj('Tag','lfacGUI'),'Tag','vrrb'),'Value',0);
	set(findobj(findobj('Tag','lfacGUI'),'Tag','lossrb'),'Value',0);
	set(findobj(findobj('Tag','lfacGUI'),'Tag','cclrb'),'Value',0);
	set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value',1);
	set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcont'),'String','1');

case 'Posslider'
	length=str2num(get(findobj(findobj('Tag','lfacGUI'),'Tag','lengthe'),'String'));
	UserData=get(findobj(findobj('Tag','lfacGUI'),'Tag','vrrb'),'UserData');
	set(findobj(findobj('Tag','lfacGUI'),'Tag','posslider'),'Max',length);
	set(findobj(findobj('Tag','lfacGUI'),'Tag','posslider'),'Sliderstep',[1/length 10/length]);
	set(findobj(findobj('Tag','lfacGUI'),'Tag','posslider'),'Min',1);
	position=round(get(findobj(findobj('Tag','lfacGUI'),'Tag','posslider'),'Value'));
	if position>length
		set(findobj(findobj('Tag','lfacGUI'),'Tag','posslider'),'Value',1);
		set(findobj(findobj('Tag','lfacGUI'),'Tag','Disppos'),'String','1');
		UserData(38)=1;
	else
		set(findobj(findobj('Tag','lfacGUI'),'Tag','Disppos'),'String',num2str(position));
		UserData(38)=round(get(findobj(findobj('Tag','lfacGUI'),'Tag','posslider'),'Value'));
	end;
	set(findobj(findobj('Tag','lfacGUI'),'Tag','vrrb'),'UserData',UserData);
	Change2dPlots([1 0 0])

case 'Yearslider'
	years=str2num(get(findobj(findobj('Tag','lfacGUI'),'Tag','yearse'),'String'));
	UserData=get(findobj(findobj('Tag','lfacGUI'),'Tag','vrrb'),'UserData');
	set(findobj(findobj('Tag','lfacGUI'),'Tag','yearslider'),'Max',years);
	set(findobj(findobj('Tag','lfacGUI'),'Tag','yearslider'),'Sliderstep',[1/years 10/years]);
	set(findobj(findobj('Tag','lfacGUI'),'Tag','yearslider'),'Min',0);
	year=round(get(findobj(findobj('Tag','lfacGUI'),'Tag','yearslider'),'Value'));
	if year>years
		set(findobj(findobj('Tag','lfacGUI'),'Tag','yearslider'),'Value',0);
		set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispyear'),'String','0');
		UserData(39)=0;
	else
		set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispyear'),'String',num2str(year));
		UserData(39)=round(get(findobj(findobj('Tag','lfacGUI'),'Tag','yearslider'),'Value'));
	end;
	set(findobj(findobj('Tag','lfacGUI'),'Tag','vrrb'),'UserData',UserData);
	Change2dPlots([0 1 0])

case 'Contourslider'
	vr=get(findobj(findobj('Tag','lfacGUI'),'Tag','vrrb'),'Value');
	loss=get(findobj(findobj('Tag','lfacGUI'),'Tag','lossrb'),'Value');
	ccl=get(findobj(findobj('Tag','lfacGUI'),'Tag','cclrb'),'Value');
	ati=get(findobj(findobj('Tag','lfacGUI'),'Tag','ATIrb'),'Value');
	if vr==1
		set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Max',100);
		set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Sliderstep',[0.005 0.05]);
		set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Min',-100);
		valuecontslider=round(get(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value'));
		if valuecontslider>100
			set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value',0);
		end;
		set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcont'),'String',num2str(valuecontslider));
	elseif loss==1
```

```
            set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Max',10e3);
            set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Sliderstep',[0.001 0.01]);
            set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Min',0);
            valuecontslider=round(get(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value'));
            if valuecontslider==0
                set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value',10);
            else
                set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcont'),'String',num2str(valuecontslider));
            end;
        elseif ccl==1
            set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Max',10e3);
            set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Sliderstep',[0.001 0.01]);
            set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Min',0);
            valuecontslider=round(get(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value'));
            if valuecontslider==0
                set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value',10);
            else;
                set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcont'),'String',num2str(valuecontslider*10000));
            end;
        elseif ati==1
            set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Max',10);
            set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Sliderstep',[0.11111111 0.5]);
            set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Min',1);
            valuecontslider=round(get(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value'));
            if valuecontslider==0
                set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value',1);
            elseif valuecontslider>10
                set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value',1);
            else;
                set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcont'),'String',num2str(valuecontslider));
            end;
        end;

    case 'r1'
        set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))          % erase input on error
    case 'r2'
        set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))          % erase input on error
    case 'r3'
        set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))          % erase input on error
    case 'x1'
        set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))          % erase input on error
    case 'x2'
        set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))          % erase input on error
    case 'x3'
        set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))          % erase input on error
    case 'b1'
        set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))          % erase input on error
    case 'b2'
        set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))          % erase input on error
    case 'b3'
        set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))          % erase input on error
    case 'cost1'
        set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))          % erase input on error
    case 'cost2'
        set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))          % erase input on error
    case 'cost3'
        set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))          % erase input on error
    case 'sysV'
        set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))          % erase input on error
    case 'Vs'
        set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))          % erase input on error
    case 'length'
        oldvalue=get(findobj(findobj('Tag','lfacGUI'),'Tag','posslider'),'Max');
        set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))          % erase input on error
        newvalue=str2num(get(gcbo,'String'));
        if newvalue<=1
            set(gcbo,'String',oldvalue);
        end;
    case 'LG'
        set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))          % erase input on error
    case 'IL'
        set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))          % erase input on error
    case 'years'
        oldvalue=get(findobj(findobj('Tag','lfacGUI'),'Tag','yearslider'),'Max');
        set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))          % erase input on error
        newvalue=str2num(get(gcbo,'String'));
        if newvalue<=0
            set(gcbo,'String',oldvalue);
        elseif newvalue>25
            set(gcbo,'String',oldvalue);
        end;
    case 'stationcost'
        set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))          % erase input on error
    case 'interest'
        set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))          % erase input on error
    case 'ALF'
        set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))          % erase input on error
    case 'ampc1'
        set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))          % erase input on error
    case 'ampc2'
        set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))          % erase input on error
    case 'ampc3'
        set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))          % erase input on error
    case 'w1'
        set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))          % erase input on error
        value=str2num(get(gcbo,'String'));
        if value>1
            set(gcbo,'String','');
        end;
    case 'w2'
        set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))          % erase input on error
```

```
		value=str2num(get(gcbo,'String'));
		if value>1
			set(gcbo,'String','');
		end;
	case 'w3'
		set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))			% erase input on error
		value=str2num(get(gcbo,'String'));
		if value>1
			set(gcbo,'String','');
		end;
	case 'w4'
		set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))			% erase input on error
		value=str2num(get(gcbo,'String'));
		if value>1
			set(gcbo,'String','');
		end;

	case 'Run'
		%Initialize variables
		clear all;
		plotc1cont=[];
		plotc2cont=[];
		plotc3cont=[];

		GC=str2num(get(findobj(findobj('Tag','lfacGUI'),'Tag','gclb'),'String'));
		namec1=get(findobj(findobj('Tag','lfacGUI'),'Tag','c1e'),'String');		%Get name of conductor 1
		namec2=get(findobj(findobj('Tag','lfacGUI'),'Tag','c2e'),'String');		%Get name of conductor 2
		namec3=get(findobj(findobj('Tag','lfacGUI'),'Tag','c3e'),'String');		%Get name of conductor 3

		UserData(1)=(str2num(get(findobj(findobj('Tag','lfacGUI'),'Tag','r1e'),'String')));
		UserData(2)=(str2num(get(findobj(findobj('Tag','lfacGUI'),'Tag','r2e'),'String')));
		UserData(3)=(str2num(get(findobj(findobj('Tag','lfacGUI'),'Tag','r3e'),'String')));
		UserData(4)=(str2num(get(findobj(findobj('Tag','lfacGUI'),'Tag','x1e'),'String')));
		UserData(5)=(str2num(get(findobj(findobj('Tag','lfacGUI'),'Tag','x2e'),'String')));
		UserData(6)=(str2num(get(findobj(findobj('Tag','lfacGUI'),'Tag','x3e'),'String')));
		UserData(7)=(str2num(get(findobj(findobj('Tag','lfacGUI'),'Tag','b1e'),'String')))*1e-6;
		UserData(8)=(str2num(get(findobj(findobj('Tag','lfacGUI'),'Tag','b2e'),'String')))*1e-6;
		UserData(9)=(str2num(get(findobj(findobj('Tag','lfacGUI'),'Tag','b3e'),'String')))*1e-6;
		UserData(10)=str2num(get(findobj(findobj('Tag','lfacGUI'),'Tag','costc1e'),'String'));
		UserData(11)=str2num(get(findobj(findobj('Tag','lfacGUI'),'Tag','costc2e'),'String'));
		UserData(12)=str2num(get(findobj(findobj('Tag','lfacGUI'),'Tag','costc3e'),'String'));
		UserData(13)=str2num(get(findobj(findobj('Tag','lfacGUI'),'Tag','ampc1e'),'String'));
		UserData(14)=str2num(get(findobj(findobj('Tag','lfacGUI'),'Tag','ampc2e'),'String'));
		UserData(15)=str2num(get(findobj(findobj('Tag','lfacGUI'),'Tag','ampc3e'),'String'));
		UserData(16)=str2num(get(findobj(findobj('Tag','lfacGUI'),'Tag','lengthe'),'String'));		%line length [km]
		UserData(17)=str2num(get(findobj(findobj('Tag','lfacGUI'),'Tag','yearse'),'String'));		%Get years
		UserData(18)=str2num(get(findobj(findobj('Tag','lfacGUI'),'Tag','ile'),'String'))*1e6;		%Get initial load
		UserData(19)=str2num(get(findobj(findobj('Tag','lfacGUI'),'Tag','lge'),'String'))/100;		%Get load growth [%]
		UserData(20)=str2num(get(findobj(findobj('Tag','lfacGUI'),'Tag','pfe'),'String'));		%Get power factor
		UserData(22)=str2num(get(findobj(findobj('Tag','lfacGUI'),'Tag','sysve'),'String'))*1e3;		%Get system voltage
		UserData(21)=(str2num(get(findobj(findobj('Tag','lfacGUI'),'Tag','sendve'),'String'))/100)*UserData(22);
		UserData(23)=(str2num(get(findobj(findobj('Tag','lfacGUI'),'Tag','inte'),'String')))/100;
		UserData(24)=(str2num(get(findobj(findobj('Tag','lfacGUI'),'Tag','alfe'),'String')))^2;
		UserData(25)=str2num(get(findobj(findobj('Tag','lfacGUI'),'Tag','ccse'),'String'));
		UserData(26)=str2num(get(findobj(findobj('Tag','lfacGUI'),'Tag','w1e'),'String'));
		UserData(27)=str2num(get(findobj(findobj('Tag','lfacGUI'),'Tag','w2e'),'String'));
		UserData(28)=str2num(get(findobj(findobj('Tag','lfacGUI'),'Tag','w3e'),'String'));
		UserData(29)=str2num(get(findobj(findobj('Tag','lfacGUI'),'Tag','w4e'),'String'));
		UserData(30)=get(findobj(findobj('Tag','lfacGUI'),'Tag','vrrb'),'Value');
		UserData(31)=get(findobj(findobj('Tag','lfacGUI'),'Tag','lossrb'),'Value');
		UserData(32)=get(findobj(findobj('Tag','lfacGUI'),'Tag','cclrb'),'Value');
		UserData(33)=get(findobj(findobj('Tag','lfacGUI'),'Tag','ATIrb'),'Value');

		UserData(35)=get(findobj(findobj('Tag','lfacGUI'),'Tag','CCC1cb'),'Value');
		UserData(36)=get(findobj(findobj('Tag','lfacGUI'),'Tag','CCC2cb'),'Value');
		UserData(37)=get(findobj(findobj('Tag','lfacGUI'),'Tag','CCC3cb'),'Value');
		SetSliders(UserData);

		UserData(34)=round(get(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value'));
		UserData(38)=round(get(findobj(findobj('Tag','lfacGUI'),'Tag','posslider'),'Value'));
		UserData(39)=round(get(findobj(findobj('Tag','lfacGUI'),'Tag','yearslider'),'Value'));

		%ac Voltage regulationl/loss/CCL/ATI functions
		[Data,XData,XData123,LR,XD,YData,plotc1cont,plotc2cont,plotc3cont]
					=VRandLoss(UserData,GC,plotc1cont,plotc2cont,plotc3cont);

		%3d plots of voltage regulation/loss/CCL/ATI
		plot3dac(XData123,YData,namec1,namec2,namec3,UserData,Data)

		%2d plots of cross-section through 3d voltage regulation/loss/CCL/ATI plots
		plot2dac(XData,XData123,LR,XD,YData,namec1,namec2,namec3,plotc1cont,plotc2cont,plotc3cont,UserData,Data)

		set(findobj(findobj('Tag','lfacGUI'),'Tag','vrrb'),'UserData',UserData);
		set(findobj(findobj('Tag','lfacGUI'),'Tag','lossrb'),'UserData',Data);
		set(findobj(findobj('Tag','lfacGUI'),'Tag','cclrb'),'UserData',XData123);
		set(findobj(findobj('Tag','lfacGUI'),'Tag','ATIrb'),'UserData',YData);
		set(findobj(findobj('Tag','lfacGUI'),'Tag','rpb'),'UserData',XD);
		set(findobj(findobj('Tag','lfacGUI'),'Tag','resetpb'),'UserData',XData);
		set(findobj(findobj('Tag','lfacGUI'),'Tag','closepb'),'UserData',LR);
		set(findobj(findobj('Tag','lfacGUI'),'Tag','c1e'),'UserData',plotc1cont);
		set(findobj(findobj('Tag','lfacGUI'),'Tag','c2e'),'UserData',plotc2cont);
		set(findobj(findobj('Tag','lfacGUI'),'Tag','c3e'),'UserData',plotc3cont);

	case 'reset'
		set(findobj(findobj('Tag','lfacGUI'),'Tag','vrrb'),'Value',1);
		set(findobj(findobj('Tag','lfacGUI'),'Tag','lossrb'),'Value',0);
		set(findobj(findobj('Tag','lfacGUI'),'Tag','cclrb'),'Value',0);
		set(findobj(findobj('Tag','lfacGUI'),'Tag','ATIrb'),'Value',0);

		set(findobj(findobj('Tag','lfacGUI'),'Tag','posslider'),'Value',1);
		set(findobj(findobj('Tag','lfacGUI'),'Tag','posslider'),'Max',10);
```

```
        set(findobj(findobj('Tag','lfacGUI'),'Tag','Disppos'),'String','1');
        set(findobj(findobj('Tag','lfacGUI'),'Tag','yearslider'),'Value',0);
        set(findobj(findobj('Tag','lfacGUI'),'Tag','yearslider'),'Max',25);
        set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispyear'),'String','0');
        set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value',1);
        set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Max',100);
        set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcont'),'String','1');

        set(findobj(findobj('Tag','lfacGUI'),'Tag','c1e'),'String','');
        set(findobj(findobj('Tag','lfacGUI'),'Tag','r1e'),'String','');
        set(findobj(findobj('Tag','lfacGUI'),'Tag','x1e'),'String','');
        set(findobj(findobj('Tag','lfacGUI'),'Tag','b1e'),'String','');
        set(findobj(findobj('Tag','lfacGUI'),'Tag','costc1e'),'String','');
        set(findobj(findobj('Tag','lfacGUI'),'Tag','ampc1e'),'String','');
        set(findobj(findobj('Tag','lfacGUI'),'Tag','c1t'),'BackgroundColor',[.8314 .8157 .7843]);
        set(findobj(findobj('Tag','lfacGUI'),'Tag','c2e'),'String','');
        set(findobj(findobj('Tag','lfacGUI'),'Tag','r2e'),'String','');
        set(findobj(findobj('Tag','lfacGUI'),'Tag','x2e'),'String','');
        set(findobj(findobj('Tag','lfacGUI'),'Tag','b2e'),'String','');
        set(findobj(findobj('Tag','lfacGUI'),'Tag','costc2e'),'String','');
        set(findobj(findobj('Tag','lfacGUI'),'Tag','ampc2e'),'String','');
        set(findobj(findobj('Tag','lfacGUI'),'Tag','c2t'),'BackgroundColor',[.8314 .8157 .7843]);
        set(findobj(findobj('Tag','lfacGUI'),'Tag','c3e'),'String','');
        set(findobj(findobj('Tag','lfacGUI'),'Tag','r3e'),'String','');
        set(findobj(findobj('Tag','lfacGUI'),'Tag','x3e'),'String','');
        set(findobj(findobj('Tag','lfacGUI'),'Tag','b3e'),'String','');
        set(findobj(findobj('Tag','lfacGUI'),'Tag','costc3e'),'String','');
        set(findobj(findobj('Tag','lfacGUI'),'Tag','ampc3e'),'String','');
        set(findobj(findobj('Tag','lfacGUI'),'Tag','c3t'),'BackgroundColor',[.8314 .8157 .7843]);
        set(findobj(findobj('Tag','lfacGUI'),'Tag','sysve'),'String','');
        set(findobj(findobj('Tag','lfacGUI'),'Tag','sendve'),'String','');
        set(findobj(findobj('Tag','lfacGUI'),'Tag','lengthe'),'String','');
        set(findobj(findobj('Tag','lfacGUI'),'Tag','lge'),'String','');
        set(findobj(findobj('Tag','lfacGUI'),'Tag','ile'),'String','');
        set(findobj(findobj('Tag','lfacGUI'),'Tag','yearse'),'String','');
        set(findobj(findobj('Tag','lfacGUI'),'Tag','inte'),'String','');
        set(findobj(findobj('Tag','lfacGUI'),'Tag','gclb'),'Value',1);
        set(findobj(findobj('Tag','lfacGUI'),'Tag','alfe'),'String','');
        set(findobj(findobj('Tag','lfacGUI'),'Tag','ccse'),'String','');
        set(findobj(findobj('Tag','lfacGUI'),'Tag','w1e'),'String','');
        set(findobj(findobj('Tag','lfacGUI'),'Tag','w2e'),'String','');
        set(findobj(findobj('Tag','lfacGUI'),'Tag','w3e'),'String','');
        set(findobj(findobj('Tag','lfacGUI'),'Tag','w4e'),'String','');
        set(findobj(findobj('Tag','lfacGUI'),'Tag','pfe'),'String','');

        axes(findobj(findobj('Tag','lfacGUI'),'Tag','vrl1axes'))
        cla reset
        set(gca,'Tag','vrl1axes');
        axes(findobj(findobj('Tag','lfacGUI'),'Tag','vrl2axes'))
        cla reset
        set(gca,'Tag','vrl2axes');
        axes(findobj(findobj('Tag','lfacGUI'),'Tag','vrl3axes'))
        cla reset
        set(gca,'Tag','vrl3axes');
        axes(findobj(findobj('Tag','lfacGUI'),'Tag','vrlposaxes'))
        cla reset
        set(gca,'Tag','vrlposaxes');
        axes(findobj(findobj('Tag','lfacGUI'),'Tag','vrlyearaxes'))
        cla reset
        set(gca,'Tag','vrlyearaxes');
        axes(findobj(findobj('Tag','lfacGUI'),'Tag','vrlcontaxes'))
        cla reset
        set(gca,'Tag','vrlcontaxes');

        if get(findobj(findobj('Tag','lfacGUI'),'Tag','hrfcb'),'Value')
            set(findobj(findobj('Tag','lfacGUI'),'Tag','hrfcb'),'Value',0)
            fig = findobj('Tag','3dvrlc1fig');
            if ~isempty(fig)
                close(figure(fig));
            end;
            fig = findobj('Tag','3dvrlc2fig');
            if ~isempty(fig)
                close(figure(fig));
            end;
            fig = findobj('Tag','3dvrlc3fig');
            if ~isempty(fig)
                close(figure(fig));
            end;
            fig = findobj('Tag','2dvrlpointloadfig');
            if ~isempty(fig)
                close(figure(fig));
            end;
            fig = findobj('Tag','2dvrlposfig');
            if ~isempty(fig)
                close(figure(fig));
            end;
            fig = findobj('Tag','2dvrlcontfig');
            if ~isempty(fig)
                close(figure(fig));
            end;
        end;

    case 'CloseHighResolutionFigures'
        fig = findobj('Tag','3dvrlc1fig');
        if ~isempty(fig)
            close(figure(fig));
        end;
        fig = findobj('Tag','3dvrlc2fig');
        if ~isempty(fig)
            close(figure(fig));
        end;
```

```
		fig = findobj('Tag','3dvrlc3fig');
		if ~isempty(fig)
			close(figure(fig));
		end;
		fig = findobj('Tag','2dvrlpointloadfig');
		if ~isempty(fig)
			close(figure(fig));
		end;
		fig = findobj('Tag','2dvrlposfig');
		if ~isempty(fig)
			close(figure(fig));
		end;
		fig = findobj('Tag','2dvrlcontfig');
		if ~isempty(fig)
			close(figure(fig));
		end;

	case 'close'
		if get(findobj(findobj('Tag','lfacGUI'),'Tag','hrfcb'),'Value')
			set(findobj(findobj('Tag','lfacGUI'),'Tag','hrfcb'),'Value',0)
			fig = findobj('Tag','3dvrlc1fig');
			if ~isempty(fig)
				close(figure(fig));
			end;
			fig = findobj('Tag','3dvrlc2fig');
			if ~isempty(fig)
				close(figure(fig));
			end;
			fig = findobj('Tag','3dvrlc3fig');
			if ~isempty(fig)
				close(figure(fig));
			end;
			fig = findobj('Tag','2dvrlpointloadfig');
			if ~isempty(fig)
				close(figure(fig));
			end;
			fig = findobj('Tag','2dvrlposfig');
			if ~isempty(fig)
				close(figure(fig));
			end;
			fig = findobj('Tag','2dvrlcontfig');
			if ~isempty(fig)
				close(figure(fig));
			end;
		end;
		fig = findobj('Tag','lfacGUI');
		saveas(figure(fig),get(fig,'FileName'),'fig')
		close(figure(fig));

function SetSliders(UserData)
	length=UserData(16);
	years=UserData(17);
	vr=UserData(30);
	loss=UserData(31);
	ccl=UserData(32);
	ati=UserData(33);

	set(findobj(findobj('Tag','lfacGUI'),'Tag','posslider'),'Max',length);
	set(findobj(findobj('Tag','lfacGUI'),'Tag','posslider'),'Sliderstep',[1/length 10/length]);
	set(findobj(findobj('Tag','lfacGUI'),'Tag','posslider'),'Min',1);
	position=round(get(findobj(findobj('Tag','lfacGUI'),'Tag','posslider'),'Value'));
	if position>length
		set(findobj(findobj('Tag','lfacGUI'),'Tag','posslider'),'Value',1);
		set(findobj(findobj('Tag','lfacGUI'),'Tag','Disppos'),'String','1');
	else
		set(findobj(findobj('Tag','lfacGUI'),'Tag','Disppos'),'String',num2str(position));
	end;

	set(findobj(findobj('Tag','lfacGUI'),'Tag','yearslider'),'Max',years);
	set(findobj(findobj('Tag','lfacGUI'),'Tag','yearslider'),'Sliderstep',[1/years 10/years]);
	set(findobj(findobj('Tag','lfacGUI'),'Tag','yearslider'),'Min',0);
	valueyearslider=round(get(findobj(findobj('Tag','lfacGUI'),'Tag','yearslider'),'Value'));
	if valueyearslider>years
		set(findobj(findobj('Tag','lfacGUI'),'Tag','yearslider'),'Value',0);
		set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispyear'),'String','0');
	else
		set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispyear'),'String',num2str(valueyearslider));
	end;

	if vr==1
		set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Max',100);
		set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Sliderstep',[0.01 0.1]);
		set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Min',-100);
		valuecontslider=round(get(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value'));
		if valuecontslider>100
			set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value',0);
			set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcont'),'String','0');
		else
			set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcont'),'String',num2str(valuecontslider));
		end;
	elseif loss==1
		set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Max',10e3);
		set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Sliderstep',[0.001 0.01]);
		set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Min',0);
		valuecontslider=round(get(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value'));
		if valuecontslider==0
			set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value',10);
			set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcont'),'String','10');
		else
			set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcont'),'String',num2str(valuecontslider));
		end;
```

```
        elseif ccl==1
                set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Max',10e3);
                set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Sliderstep',[0.001 0.01]);
                set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Min',0);
                valuecontslider=round(get(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value'));
                if valuecontslider==0
                        set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value',10);
                        set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcont'),'String','100000');
                else
                        set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcont'),'String',num2str(valuecontslider*100000));
                end;
        elseif ati==1
                set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Max',10);
                set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Sliderstep',[0.11111111 0.5]);
                set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Min',1);
                valuecontslider=round(get(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value'));
                if valuecontslider==0
                        set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value',1);
                elseif valuecontslider>10
                        set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value',1);
                else;
                        set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcont'),'String',num2str(valuecontslider));
                end;
        end;

function [Data,XData,XData123,LR,XD,YData,plotc1cont,plotc2cont,plotc3cont]
                                =VRandLoss(UserData,GC,plotc1cont,plotc2cont,plotc3cont);
        rc1=UserData(1);
        rc2=UserData(2);
        rc3=UserData(3);
        xc1=UserData(4);
        xc2=UserData(5);
        xc3=UserData(6);
        bc1=UserData(7);
        bc2=UserData(8);
        bc3=UserData(9);
        costc1=UserData(10);
        costc2=UserData(11);
        costc3=UserData(12);
        ampc1=UserData(13);
        ampc2=UserData(14);
        ampc3=UserData(15);
        length=UserData(16);
        years=UserData(17);
        IL=UserData(18);
        LG=UserData(19);
        pf=UserData(20);
        Vs=UserData(21);
        sysV=UserData(22);
        Interest=UserData(23);
        ELF=UserData(24);
        CCS=UserData(25);
        w1=UserData(26);
        w2=UserData(27);
        w3=UserData(28);
        w4=UserData(29);
        vr=UserData(30);
        loss=UserData(31);
        ccl=UserData(32);
        ati=UserData(33);
        contour=UserData(34);
        comp1=UserData(35);
        comp2=UserData(36);
        comp3=UserData(37);

        Sbase=100e6;                                    %Power base
        Zbase=(sysV^2)/Sbase;                           %Calculate Zbase
        Vspu=Vs/sysV;                                   %Calculate the sending voltage in p.u.

        for c=1:(years+1)
                XData(c)=(IL/1e6)*(1+LG)^(c-1);         %Set up XData vector for 3d plot
                DF(c)=[(1+Interest)^(c-1)];             %Discount Factor
                CLA(c)=8760*(GC(c)/100)*ELF;            %Cost of losses/kW/Annum [R]
        end;

        XD1=zeros([length,1]);
        XD2=zeros([length,1]);
        XD3=zeros([length,1]);

        testLR1=zeros([(years+1),1]);
        testLR2=zeros([(years+1),1]);
        testLR3=zeros([(years+1),1]);

        contourmax1=0;
        contourmax2=0;
        contourmax3=0;
        contourmin1=10;
        contourmin2=10;
        contourmin3=10;

        SILlower=0;
        SILupper=0;

        yc1=bc1*j;
        Zc1=sqrt((rc1+xc1*j)/yc1);
        lamda1=sqrt((rc1+xc1*j)*yc1);

        yc2=bc2*j;
        Zc2=sqrt((rc2+xc2*j)/yc2);
        lamda2=sqrt((rc2+xc2*j)*yc2);
```

```
yc3=bc3*j;
Zc3=sqrt((rc3+xc3*j)/yc3);
lamda3=sqrt((rc3+xc3*j)*yc3);

YData=1:length;
for j=1:(years+1)
        LR1(j)=length;
        LR2(j)=length;
        LR3(j)=length;
end;

h = waitbar(0,'Calculating matrices...');

for i=1:length
        waitbar(i/length);

        Z1=Zc1*sinh(lamda1*i);
        Y1=(2*tanh((lamda1*i)/2))/Zc1;
        R1pu=real(Z1)/Zbase;
        X1pu=imag(Z1)/Zbase;
        B1pu=imag(Y1)*Zbase;

        Z2=Zc2*sinh(lamda2*i);
        Y2=(2*tanh((lamda2*i)/2))/Zc2;
        R2pu=real(Z2)/Zbase;
        X2pu=imag(Z2)/Zbase;
        B2pu=imag(Y2)*Zbase;

        Z3=Zc3*sinh(lamda3*i);
        Y3=(2*tanh((lamda3*i)/2))/Zc3;
        R3pu=real(Z3)/Zbase;
        X3pu=imag(Z3)/Zbase;
        B3pu=imag(Y3)*Zbase;

        Z1sqpu=R1pu^2+X1pu^2;
        Z2sqpu=R2pu^2+X2pu^2;
        Z3sqpu=R3pu^2+X3pu^2;

        K1=1-B1pu*X1pu+(B1pu^2*Z1sqpu)/4;
        K2=1-B2pu*X2pu+(B2pu^2*Z2sqpu)/4;
        K3=1-B3pu*X3pu+(B3pu^2*Z3sqpu)/4;

        XData1(i,:)=XData;
        XData2(i,:)=XData;
        XData3(i,:)=XData;

        for j=1:(years+1)
                Srpu=(IL/Sbase)*(1+LG)^(j-1);
                Prpu=((IL*pf)/Sbase)*(1+LG)^(j-1);
                Qrpu=sqrt(Srpu^2-Prpu^2);

                F1=(Vspu^2)-2*R1pu*Prpu-Qrpu*(2*X1pu-B1pu*Z1sqpu);
                F2=(Vspu^2)-2*R2pu*Prpu-Qrpu*(2*X2pu-B2pu*Z2sqpu);
                F3=(Vspu^2)-2*R3pu*Prpu-Qrpu*(2*X3pu-B3pu*Z3sqpu);

                Vr1pu(i,j)=sqrt((F1+sqrt(F1^2-4*Z1sqpu*(Prpu^2+Qrpu^2)*K1))/(2*K1));
                Vr2pu(i,j)=sqrt((F2+sqrt(F2^2-4*Z2sqpu*(Prpu^2+Qrpu^2)*K2))/(2*K2));
                Vr3pu(i,j)=sqrt((F3+sqrt(F3^2-4*Z3sqpu*(Prpu^2+Qrpu^2)*K3))/(2*K3));

                if isreal(Vr1pu(i,j))
                        Vreg1(i,j)=(Vspu-abs(Vr1pu(i,j)))*100;
                        Ploss1(i,j)=((R1pu/Z1sqpu)*(Vspu^2-Vr1pu(i,j)^2-
                                2*(R1pu*Prpu+X1pu*Qrpu)+B1pu*X1pu*Vr1pu(i,j)^2))*1E5;
                        ECL1(i,j)=Ploss1(i,j)*CLA(j);
                        PVCL1(i,j)=ECL1(i,j)/DF(j);
                        PVCL1(:,1)=0;
                        jrepeat1=j;
                else
                        if testLR1(j)==0
                                LR1(j)=i-1;
                                testLR1(j)=1;
                        end;
                        Vreg1(i,j)=Vreg1(i,jrepeat1);
                        Ploss1(i,j)=Ploss1(i,jrepeat1);
                        PVCL1(i,j)=0;
                        XData1(i,j)=XData(jrepeat1);
                        XD1(i)=1;
                end;

                if isreal(Vr2pu(i,j))
                        Vreg2(i,j)=(Vspu-abs(Vr2pu(i,j)))*100;
                        Ploss2(i,j)=((R2pu/Z2sqpu)*(Vspu^2-Vr2pu(i,j)^2-
                                2*(R2pu*Prpu+X2pu*Qrpu)+B2pu*X2pu*Vr2pu(i,j)^2))*1E5;
                        ECL2(i,j)=Ploss2(i,j)*CLA(j);
                        PVCL2(i,j)=ECL2(i,j)/DF(j);
                        PVCL2(:,1)=0;
                        jrepeat2=j;
                else
                        if testLR2(j)==0
                                LR2(j)=i-1;
                                testLR2(j)=1;
                        end;
                        Vreg2(i,j)=Vreg2(i,jrepeat2);
                        Ploss2(i,j)=Ploss2(i,jrepeat2);
                        PVCL2(i,j)=0;
                        XData2(i,j)=XData(jrepeat2);
                        XD2(i)=1;
                end;

                if isreal(Vr3pu(i,j))
                        Vreg3(i,j)=(Vspu-abs(Vr3pu(i,j)))*100;
```

```
                    Ploss3(i,j)=((R3pu/Z3sqpu)*(Vspu^2-Vr3pu(i,j)^2-
                                2*(R3pu*Prpu+X3pu*Qrpu)+B3pu*X3pu*Vr3pu(i,j)^2))*1E5;
                    ECL3(i,j)=Ploss3(i,j)*CLA(j);
                    PVCL3(i,j)=ECL3(i,j)/DF(j);
                    PVCL3(:,1)=0;
                    jrepeat3=j;
                else
                    if testLR3(j)==0
                        LR3(j)=i-1;
                        testLR3(j)=1;
                    end;
                    Vreg3(i,j)=Vreg3(i,jrepeat3);
                    Ploss3(i,j)=Ploss3(i,jrepeat3);
                    PVCL3(i,j)=0;
                    XData3(i,j)=XData(jrepeat3);
                    XD3(i)=1;
                end;

                if (vr==1) & (contour>0) & (Vreg1(i,j)>=contour)
                    plotc1cont=1;
                elseif (vr==1) & (contour<=0) & (Vreg1(i,j)<=contour)
                    plotc1cont=1;
                elseif (loss==1) & (Ploss1(i,j)>=contour)
                    plotc1cont=1;
                end;

                if (vr==1) & (contour>0) & (Vreg2(i,j)>=contour)
                    plotc2cont=1;
                elseif (vr==1) & (contour<=0) & (Vreg2(i,j)<=contour)
                    plotc2cont=1;
                elseif (loss==1) & (Ploss2(i,j)>=contour)
                    plotc2cont=1;
                end;

                if (vr==1) & (contour>0) & (Vreg3(i,j)>=contour)
                    plotc3cont=1;
                elseif (vr==1) & (contour<=0) & (Vreg3(i,j)<=contour)
                    plotc3cont=1;
                elseif (loss==1) & (Ploss3(i,j)>=contour)
                    plotc3cont=1;
                end;

                Capacityc1(i,j)=(costc1*i)/(sqrt(3)*sysV*1e-3*ampc1*1*i);        %Capacity Matrix of conductor 1
                Capacityc2(i,j)=(costc2*i)/(sqrt(3)*sysV*1e-3*ampc2*1*i);        %Capacity Matrix of conductor 2
                Capacityc3(i,j)=(costc3*i)/(sqrt(3)*sysV*1e-3*ampc3*1*i);        %Capacity Matrix of conductor 3

                SILc1(i,j)=(costc1*sqrt(abs(Z1)/imag(Y1)))/((sysV/1e3)^2);
                SILc2(i,j)=(costc2*sqrt(abs(Z2)/imag(Y2)))/((sysV/1e3)^2);
                SILc3(i,j)=(costc3*sqrt(abs(Z3)/imag(Y3)))/((sysV/1e3)^2);

            end;

            CPVCL1(i,:)=cumsum(PVCL1(i,:));        %Cumulative Present Value Cost of Losses for Conductor 1
            CPVCL2(i,:)=cumsum(PVCL2(i,:));        %Cumulative Present Value Cost of Losses for Conductor 2
            CPVCL3(i,:)=cumsum(PVCL3(i,:));        %Cumulative Present Value Cost of Losses for Conductor 3

            if comp1==1
                PVCCL1(i,:)=(CPVCL1(i,:)+costc1*i+CCS)/1e6;
            else
                PVCCL1(i,:)=(CPVCL1(i,:)+costc1*i)/1e6;
            end;
            if comp2==1
                PVCCL2(i,:)=(CPVCL2(i,:)+costc2*i+CCS)/1e6;
            else
                PVCCL2(i,:)=(CPVCL2(i,:)+costc2*i)/1e6;
            end;
            if comp3==1
                PVCCL3(i,:)=(CPVCL3(i,:)+costc3*i+CCS)/1e6;
            else
                PVCCL3(i,:)=(CPVCL3(i,:)+costc3*i)/1e6;
            end;

            if SILc1(i,:)>SILlower
                SILlower=SILc1(i,1);
            elseif SILc2(i,:)>SILlower
                SILlower=SILc2(i,1);
            elseif SILc3(i,:)>SILlower
                SILlower=SILc3(i,1);
            end;
            if SILc1(i,:)<SILc2(i,:) & SILc1(i,:)<SILc3(i,:)
                SILupper=SILc1(i,1);
            elseif SILc2(i,:)<SILc3(i,:)
                SILupper=SILc2(i,1);
            else
                SILupper=SILc3(i,1);
            end;

            for n=1:(years+1)
                if (ccl==1) & (PVCCL1(i,n)>=contour*10000) & (contour*10000>PVCCL1(1,1))
                    plotc1cont=1;
                end;
                if (ccl==1) & (PVCCL2(i,n)>=contour*10000) & (contour*10000>PVCCL2(1,1))
                    plotc2cont=1;
                end;
                if (ccl==1) & (PVCCL3(i,n)>=contour*10000) & (contour*10000>PVCCL3(1,1))
                    plotc3cont=1;
                end;
            end;

        end;
```

```
        close(h);
        if isreal(Vr1pu)
                set(findobj(findobj('Tag','lfacGUI'),'Tag','c1t'),'BackgroundColor',[.8314 .8157 .7843]);
        else
                set(findobj(findobj('Tag','lfacGUI'),'Tag','c1t'),'BackgroundColor','r');
        end;
        if isreal(Vr2pu)
                set(findobj(findobj('Tag','lfacGUI'),'Tag','c2t'),'BackgroundColor',[.8314 .8157 .7843]);
        else
                set(findobj(findobj('Tag','lfacGUI'),'Tag','c2t'),'BackgroundColor','r');
        end;
        if isreal(Vr3pu)
                set(findobj(findobj('Tag','lfacGUI'),'Tag','c3t'),'BackgroundColor',[.8314 .8157 .7843]);
        else
                set(findobj(findobj('Tag','lfacGUI'),'Tag','c3t'),'BackgroundColor','r');
        end;

        %Calculate upper, median and lower values for LCC KPI
        LCClower=0;
        LCCupper=0;
        if PVCCL1(length,years+1)>PVCCL2(length,years+1) & PVCCL1(length,years+1)>PVCCL3(length,years+1)
                LCClower=PVCCL1(length,years+1)+1;
        elseif PVCCL2(length,years+1)>PVCCL3(length,years+1)
                LCClower=PVCCL2(length,years+1)+1;
        else
                LCClower=PVCCL3(length,years+1)+1;
        end;
        LCCmedian=LCClower-(((LCClower-LCCupper)/10)*3);

        %Calculate upper, median and lower values for Capacity KPI
        Capacitylower=0;
        Capacityupper=0;
        if Capacityc1(1,1)<Capacityc2(1,1) & Capacityc1(1,1)<Capacityc3(1,1)
                Capacityupper=Capacityc1(1,1)-1;
        elseif Capacityc2(1,1)<Capacityc3(1,1)
                Capacityupper=Capacityc2(1,1)-1;
        else
                Capacityupper=Capacityc3(1,1)-1;
        end;

        if Capacityc1(1,1)>Capacityc2(1,1) & Capacityc1(1,1)>Capacityc3(1,1)
                Capacitylower=Capacityc1(1,1)+1;
        elseif Capacityc2(1,1)>Capacityc3(1,1)
                Capacitylower=Capacityc2(1,1)+1;
        else;
                Capacitylower=Capacityc3(1,1)+1;
        end;
        Capacitymedian=Capacitylower-(((Capacitylower-Capacityupper)/10)*3);

        %Calculate median values for SIL KPI
        SILmedian=SILlower-(((SILlower-SILupper)/10)*3);

        %Calculate upper, median and lower values for Voltage Regulation KPI
        VRupper=0;
        VRlower=0.50;
        VRmedian=VRlower-(((VRlower-VRupper)/10)*3);

        h = waitbar(0,'Calculating ATIs...');

        for i=1:length
                waitbar(i/length)
                for j=1:years+1
                        LCCc1(i,j)=10-(PVCCL1(i,j)-LCCupper)/((LCClower-LCCupper)/10);
                        LCCc2(i,j)=10-(PVCCL2(i,j)-LCCupper)/((LCClower-LCCupper)/10);
                        LCCc3(i,j)=10-(PVCCL3(i,j)-LCCupper)/((LCClower-LCCupper)/10);

                        Capacityc1(i,j)=10-(Capacityc1(i,j)-Capacityupper)/((Capacitylower-Capacityupper)/10);
                        Capacityc2(i,j)=10-(Capacityc2(i,j)-Capacityupper)/((Capacitylower-Capacityupper)/10);
                        Capacityc3(i,j)=10-(Capacityc3(i,j)-Capacityupper)/((Capacitylower-Capacityupper)/10);

                        SILc1(i,j)=10-(SILc1(i,j)-SILupper)/((SILlower-SILupper)/10);
                        SILc2(i,j)=10-(SILc2(i,j)-SILupper)/((SILlower-SILupper)/10);
                        SILc3(i,j)=10-(SILc3(i,j)-SILupper)/((SILlower-SILupper)/10);

                        %Especially only for shunt compensation
                        if comp1==1
                                VRc1(i,j)=10;
                        else
                                VRc1(i,j)=10-((abs(Vreg1(i,j)/100))-VRupper)/((VRlower-VRupper)/10);
                        end;
                        if comp2==1
                                VRc2(i,j)=10;
                        else
                                VRc2(i,j)=10-((abs(Vreg2(i,j)/100))-VRupper)/((VRlower-VRupper)/10);
                        end;
                        if comp3==1
                                VRc3(i,j)=10;
                        else
                                VRc3(i,j)=10-((abs(Vreg3(i,j)/100)-VRupper))/((VRlower-VRupper)/10);
                        end;

                        ATI1(i,j)=LCCc1(i,j)*w1 + Capacityc1(i,j)*w2 + SILc1(i,j)*w3 + VRc1(i,j)*w4;
                        ATI2(i,j)=LCCc2(i,j)*w1 + Capacityc2(i,j)*w2 + SILc2(i,j)*w3 + VRc2(i,j)*w4;
                        ATI3(i,j)=LCCc3(i,j)*w1 + Capacityc3(i,j)*w2 + SILc3(i,j)*w3 + VRc3(i,j)*w4;

                        %Determine minimum and maximum values for ATI contour plot
                        if (ati==1) & (ATI1(i,j)>contourmax1)
                                contourmax1=ATI1(i,j);
                        end;
                        if (ati==1) & (ATI2(i,j)>contourmax2)
                                contourmax2=ATI2(i,j);
```

```
                    end;
                    if (ati==1) & (ATI3(i,j)>contourmax3)
                            contourmax3=ATI3(i,j);
                    end;

                    if (ati==1) & (ATI1(i,j)<contourmin1)
                            contourmin1=ATI1(i,j);
                    end;
                    if (ati==1) & (ATI2(i,j)<contourmin2)
                            contourmin2=ATI2(i,j);
                    end;
                    if (ati==1) & (ATI3(i,j)<contourmin3)
                            contourmin3=ATI3(i,j);
                    end;

            end;
    end;

    close(h);

    %Determine if contour plot is valid for ATIs of conductor 1,2 and 3
    if (ati==1) & (contour>=contourmin1) & (contour<=contourmax1)
            plotc1cont=1;
    end;
    if (ati==1) & (contour>=contourmin2) & (contour<=contourmax2)
            plotc2cont=1;
    end;
    if (ati==1) & (contour>=contourmin3) & (contour<=contourmax3)
            plotc3cont=1;
    end;

    Data(:,:,1) = Vreg1;
    Data(:,:,2) = Vreg2;
    Data(:,:,3) = Vreg3;
    Data(:,:,4) = Ploss1;
    Data(:,:,5) = Ploss2;
    Data(:,:,6) = Ploss3;
    Data(:,:,7) = PVCCL1;
    Data(:,:,8) = PVCCL2;
    Data(:,:,9) = PVCCL3;
    Data(:,:,10) = ATI1;
    Data(:,:,11) = ATI2;
    Data(:,:,12) = ATI3;

    XData123(:,:,1) = XData1;
    XData123(:,:,2) = XData2;
    XData123(:,:,3) = XData3;

    XD(:,:,1) =     XD1;
    XD(:,:,2) =     XD2;
    XD(:,:,3) =     XD3;

    LR(:,:,1) =     LR1;
    LR(:,:,2) =     LR2;
    LR(:,:,3) =     LR3;
```

B.4.2 LFDC.m

```
function lfdc(action)
%Main Loadflow for dc bipole line GUI

switch(action)

case 'VoltageRegulation'
        set(gcbo,'Value',1);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcontunit'),'String','%');
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','lossrb'),'Value',0);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','cclrb'),'Value',0);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','ATIrb'),'Value',0);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value',1);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcont'),'String','1');

case 'Loss'
        set(gcbo,'Value',1);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcontunit'),'String','kW');
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','vrrb'),'Value',0);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','cclrb'),'Value',0);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','ATIrb'),'Value',0);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value',10);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcont'),'String','10');

case 'CCL'
        set(gcbo,'Value',1);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcontunit'),'String','R');
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','vrrb'),'Value',0);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','lossrb'),'Value',0);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','ATIrb'),'Value',0);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value',10);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcont'),'String','100000');

case 'ATI'
        set(gcbo,'Value',1);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcontunit'),'String','');
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','vrrb'),'Value',0);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','lossrb'),'Value',0);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','cclrb'),'Value',0);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value',1);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcont'),'String','1');

case 'Posslider'
        length=str2num(get(findobj(findobj('Tag','lfdcGUI'),'Tag','lengthe'),'String'));
        UserData=get(findobj(findobj('Tag','lfdcGUI'),'Tag','vrrb'),'UserData');
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','posslider'),'Max',length);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','posslider'),'Sliderstep',[1/length 10/length]);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','posslider'),'Min',1);
        position=round(get(findobj(findobj('Tag','lfdcGUI'),'Tag','posslider'),'Value'));
        if position>length
                set(findobj(findobj('Tag','lfdcGUI'),'Tag','posslider'),'Value',1);
                set(findobj(findobj('Tag','lfdcGUI'),'Tag','Disppos'),'String','1');
                UserData(27)=1;
        else
                set(findobj(findobj('Tag','lfdcGUI'),'Tag','Disppos'),'String',num2str(position));
                UserData(27)=round(get(findobj(findobj('Tag','lfdcGUI'),'Tag','posslider'),'Value'));
        end;
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','vrrb'),'UserData',UserData);
        Change2dPlotsDC([1 0 0])

case 'Yearslider'
        years=str2num(get(findobj(findobj('Tag','lfdcGUI'),'Tag','yearse'),'String'));
        UserData=get(findobj(findobj('Tag','lfdcGUI'),'Tag','vrrb'),'UserData');
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','yearslider'),'Max',years);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','yearslider'),'Sliderstep',[1/years 10/years]);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','yearslider'),'Min',0);
        year=round(get(findobj(findobj('Tag','lfdcGUI'),'Tag','yearslider'),'Value'));
        if year>years
                set(findobj(findobj('Tag','lfdcGUI'),'Tag','yearslider'),'Value',0);
                set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispyear'),'String','0');
                UserData(28)=0;
        else
                set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispyear'),'String',num2str(year));
                UserData(28)=round(get(findobj(findobj('Tag','lfdcGUI'),'Tag','yearslider'),'Value'));
        end;
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','vrrb'),'UserData',UserData);
        Change2dPlotsDC([0 1 0])

case 'Contourslider'
        vr=get(findobj(findobj('Tag','lfdcGUI'),'Tag','vrrb'),'Value');
        loss=get(findobj(findobj('Tag','lfdcGUI'),'Tag','lossrb'),'Value');
        ccl=get(findobj(findobj('Tag','lfdcGUI'),'Tag','cclrb'),'Value');
        ati=get(findobj(findobj('Tag','lfdcGUI'),'Tag','ATIrb'),'Value');
        if vr==1
                set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Max',100);
                set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Sliderstep',[0.01 0.1]);
                set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Min',1);
                valuecontslider=round(get(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value'));
                if valuecontslider>100
                        set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value',1);
                end;
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcont'),'String',num2str(valuecontslider));
        elseif loss==1
                set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Max',10e3);
                set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Sliderstep',[0.001 0.01]);
                set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Min',0);
                valuecontslider=round(get(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value'));
```

```
                    if valuecontslider==0
                        set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value',10);
                    else
                        set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcont'),'String',num2str(valuecontslider));
                    end;
                elseif ccl==1
                    set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Max',10e3);
                    set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Sliderstep',[0.001 0.01]);
                    set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Min',0);
                    valuecontslider=round(get(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value'));
                    if valuecontslider==0
                        set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value',10);
                    else;
                        set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcont'),'String',num2str(valuecontslider*10000));
                    end;
                elseif ati==1
                    set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Max',10);
                    set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Sliderstep',[0.11111111 0.5]);
                    set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Min',1);
                    valuecontslider=round(get(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value'));
                    if valuecontslider==0
                        set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value',1);
                    elseif valuecontslider>10
                        set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value',1);
                    else
                        set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcont'),'String',num2str(valuecontslider));
                    end;
                end;

        case 'r1'
            set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))                    %erase input on error
        case 'r2'
            set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))                    %erase input on error
        case 'r3'
            set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))                    %erase input on error
        case 'sysV'
            set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))                    %erase input on error
        case 'Vs'
            set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))                    %erase input on error
        case 'length'
            oldvalue=get(findobj(findobj('Tag','lfdcGUI'),'Tag','posslider'),'Max');
            set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))                    %erase input on error
            newvalue=str2num(get(gcbo,'String'));
            if newvalue<=1
                set(gcbo,'String',oldvalue);
            end;
        case 'LG'
            set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))                    %erase input on error
        case 'IL'
            set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))                    %erase input on error
        case 'years'
            oldvalue=get(findobj(findobj('Tag','lfdcGUI'),'Tag','yearslider'),'Max');
            set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))                    %erase input on error
            newvalue=str2num(get(gcbo,'String'));
            if newvalue<=0
                set(gcbo,'String',oldvalue);
            elseif newvalue>25
                set(gcbo,'String',oldvalue);
            end;
        case 'stationcost'
            set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))                    %erase input on error
        case 'stationloss'
            set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))                    %erase input on error
        case 'interest'
            set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))                    %erase input on error
        case 'ALF'
            set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))                    %erase input on error
        case 'ampc1'
            set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))                    %erase input on error
        case 'ampc2'
            set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))                    %erase input on error
        case 'ampc3'
            set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))                    %erase input on error
        case 'w1'
            set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))                    %erase input on error
            value=str2num(get(gcbo,'String'));
            if value>1
                set(gcbo,'String','');
            end;
        case 'w2'
            set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))                    %erase input on error
            value=str2num(get(gcbo,'String'));
            if value>1
            set(gcbo,'String','');
            end;
        case 'w4'
            set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))                    %erase input on error
            value=str2num(get(gcbo,'String'));
            if value>1
                set(gcbo,'String','');
            end;

        case 'Run'
            %Initialize variables
            clear all;
            plotc1cont=[];
            plotc2cont=[];
            plotc3cont=[];

            GC=str2num(get(findobj(findobj('Tag','lfdcGUI'),'Tag','gclb'),'String'));
            namec1=get(findobj(findobj('Tag','lfdcGUI'),'Tag','c1e'),'String');       %Get name of conductor 1
```

```
        namec2=get(findobj(findobj('Tag','lfdcGUI'),'Tag','c2e'),'String');                    %Get name of conductor 2
        namec3=get(findobj(findobj('Tag','lfdcGUI'),'Tag','c3e'),'String');                    %Get name of conductor 3

        UserData(1)=str2num(get(findobj(findobj('Tag','lfdcGUI'),'Tag','r1e'),'String'));   %resistance/km of conductor 1
        UserData(2)=str2num(get(findobj(findobj('Tag','lfdcGUI'),'Tag','r2e'),'String'));   %resistance/km of conductor 2
        UserData(3)=str2num(get(findobj(findobj('Tag','lfdcGUI'),'Tag','r3e'),'String'));   %resistance/km of conductor 3
        UserData(4)=str2num(get(findobj(findobj('Tag','lfdcGUI'),'Tag','costc1e'),'String'));
        UserData(5)=str2num(get(findobj(findobj('Tag','lfdcGUI'),'Tag','costc2e'),'String'));
        UserData(6)=str2num(get(findobj(findobj('Tag','lfdcGUI'),'Tag','costc3e'),'String'));
        UserData(7)=str2num(get(findobj(findobj('Tag','lfdcGUI'),'Tag','ampc1e'),'String'));
        UserData(8)=str2num(get(findobj(findobj('Tag','lfdcGUI'),'Tag','ampc2e'),'String'));
        UserData(9)=str2num(get(findobj(findobj('Tag','lfdcGUI'),'Tag','ampc3e'),'String'));
        UserData(10)=str2num(get(findobj(findobj('Tag','lfdcGUI'),'Tag','sysve'),'String'))*1e3;  %Get system voltage [kV]
        UserData(11)=(str2num(get(findobj(findobj('Tag','lfdcGUI'),'Tag','sendve'),'String'))/100)*UserData(10);
        UserData(12)=str2num(get(findobj(findobj('Tag','lfdcGUI'),'Tag','lengthe'),'String'));     %Get line length [km]
        UserData(13)=str2num(get(findobj(findobj('Tag','lfdcGUI'),'Tag','lge'),'String'))/100;     %Get load growth [%]
        UserData(14)=str2num(get(findobj(findobj('Tag','lfdcGUI'),'Tag','ile'),'String'))*1e6;     %Get initial load [MW]
        UserData(15)=str2num(get(findobj(findobj('Tag','lfdcGUI'),'Tag','yearse'),'String'));      %Get years - normally 25
        UserData(16)=(str2num(get(findobj(findobj('Tag','lfdcGUI'),'Tag','inte'),'String')))/100;
        UserData(17)=(str2num(get(findobj(findobj('Tag','lfdcGUI'),'Tag','alfe'),'String')))^2;
        UserData(18)=str2num(get(findobj(findobj('Tag','lfdcGUI'),'Tag','ccse'),'String'));
        UserData(19)=(str2num(get(findobj(findobj('Tag','lfdcGUI'),'Tag','cle'),'String')))/100;
        UserData(20)=str2num(get(findobj(findobj('Tag','lfdcGUI'),'Tag','w1e'),'String'));
        UserData(21)=str2num(get(findobj(findobj('Tag','lfdcGUI'),'Tag','w2e'),'String'));
        UserData(22)=str2num(get(findobj(findobj('Tag','lfdcGUI'),'Tag','w4e'),'String'));
        UserData(23)=get(findobj(findobj('Tag','lfdcGUI'),'Tag','vrrb'),'Value');
        UserData(24)=get(findobj(findobj('Tag','lfdcGUI'),'Tag','lossrb'),'Value');
        UserData(25)=get(findobj(findobj('Tag','lfdcGUI'),'Tag','cclrb'),'Value');
        UserData(26)=get(findobj(findobj('Tag','lfdcGUI'),'Tag','ATIrb'),'Value');

        SetSliders(UserData);

        UserData(27)=round(get(findobj(findobj('Tag','lfdcGUI'),'Tag','posslider'),'Value'));
        UserData(28)=round(get(findobj(findobj('Tag','lfdcGUI'),'Tag','yearslider'),'Value'));
        UserData(29)=round(get(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value'));

        %Calculate voltage regulation,loss, ccl and ati matrices
        [Data,XData,XData123,LR,XD,YData,plotc1cont,plotc2cont,plotc3cont]
                                    =VRandLoss(UserData,GC,plotc1cont,plotc2cont,plotc3cont);

        %3d plots of voltage regulation, loss and ati plots
        plot3d(XData123,YData,namec1,namec2,namec3,UserData,Data)

        %2d plots of cross-section through 3d voltage regulation, loss and ati plots
        plot2d(XData,XData123,LR,XD,YData,namec1,namec2,namec3,plotc1cont,plotc2cont,plotc3cont,UserData,Data)

        set(findobj(findobj('Tag','lfdcGUI'),'Tag','vrrb'),'UserData',UserData);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','lossrb'),'UserData',Data);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','cclrb'),'UserData',XData123);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','ATIrb'),'UserData',YData);

        set(findobj(findobj('Tag','lfdcGUI'),'Tag','rpb'),'UserData',XD);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','resetpb'),'UserData',XData);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','closepb'),'UserData',LR);

        set(findobj(findobj('Tag','lfdcGUI'),'Tag','c1e'),'UserData',plotc1cont);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','c2e'),'UserData',plotc2cont);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','c3e'),'UserData',plotc3cont);

    case 'reset'
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','vrrb'),'Value',1);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','lossrb'),'Value',0);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','cclrb'),'Value',0);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','ATIrb'),'Value',0);

        set(findobj(findobj('Tag','lfdcGUI'),'Tag','posslider'),'Value',1);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','posslider'),'Max',10);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','Disppos'),'String','1');
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','yearslider'),'Value',0);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','yearslider'),'Max',25);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispyear'),'String','0');
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value',1);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Max',100);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcont'),'String','1');

        set(findobj(findobj('Tag','lfdcGUI'),'Tag','c1e'),'String','');
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','r1e'),'String','');
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','costc1e'),'String','');
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','ampc1e'),'String','');
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','c1t'),'BackgroundColor',[.8314 .8157 .7843]);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','c2e'),'String','');
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','r2e'),'String','');
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','costc2e'),'String','');
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','ampc2e'),'String','');
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','c2t'),'BackgroundColor',[.8314 .8157 .7843]);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','c3e'),'String','');
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','r3e'),'String','');
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','costc3e'),'String','');
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','ampc3e'),'String','');
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','c3t'),'BackgroundColor',[.8314 .8157 .7843]);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','sysve'),'String','');
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','sendve'),'String','');
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','lengthe'),'String','');
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','lge'),'String','');
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','ile'),'String','');
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','yearse'),'String','');
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','inte'),'String','');
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','gclb'),'Value',1);
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','alfe'),'String','');
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','ccse'),'String','');
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','w1e'),'String','');
```

```
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','w2e'),'String','');
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','w4e'),'String','');
        set(findobj(findobj('Tag','lfdcGUI'),'Tag','cle'),'String','');

        axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrl1axes'))
        cla reset
        set(gca,'Tag','vrl1axes');
        axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrl2axes'))
        cla reset
        set(gca,'Tag','vrl2axes');
        axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrl3axes'))
        cla reset
        set(gca,'Tag','vrl3axes');
        axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrlposaxes'))
        cla reset
        set(gca,'Tag','vrlposaxes');
        axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrlyearaxes'))
        cla reset
        set(gca,'Tag','vrlyearaxes');
        axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrlcontaxes'))
        cla reset
        set(gca,'Tag','vrlcontaxes');

        if get(findobj(findobj('Tag','lfdcGUI'),'Tag','hrfcb'),'Value')
            set(findobj(findobj('Tag','lfdcGUI'),'Tag','hrfcb'),'Value',0)
            fig = findobj('Tag','3dvrlc1fig');
            if ~isempty(fig)
                close(figure(fig));
            end;
            fig = findobj('Tag','3dvrlc2fig');
            if ~isempty(fig)
                close(figure(fig));
            end;
            fig = findobj('Tag','3dvrlc3fig');
            if ~isempty(fig)
                close(figure(fig));
            end;
            fig = findobj('Tag','2dvrlpointloadfig');
            if ~isempty(fig)
                close(figure(fig));
            end;
            fig = findobj('Tag','2dvrlposfig');
            if ~isempty(fig)
                close(figure(fig));
            end;
            fig = findobj('Tag','2dvrlcontfig');
            if ~isempty(fig)
                close(figure(fig));
            end;
        end;

    case 'CloseHighResolutionFigures'
        fig = findobj('Tag','3dvrlc1fig');
        if ~isempty(fig)
            close(figure(fig));
        end;
        fig = findobj('Tag','3dvrlc2fig');
        if ~isempty(fig)
            close(figure(fig));
        end;
        fig = findobj('Tag','3dvrlc3fig');
        if ~isempty(fig)
            close(figure(fig));
        end;
        fig = findobj('Tag','2dvrlpointloadfig');
        if ~isempty(fig)
            close(figure(fig));
        end;
        fig = findobj('Tag','2dvrlposfig');
        if ~isempty(fig)
            close(figure(fig));
        end;
        fig = findobj('Tag','2dvrlcontfig');
        if ~isempty(fig)
            close(figure(fig));
        end;

    case 'close'
        if get(findobj(findobj('Tag','lfdcGUI'),'Tag','hrfcb'),'Value')
            set(findobj(findobj('Tag','lfdcGUI'),'Tag','hrfcb'),'Value',0)
            fig = findobj('Tag','3dvrlc1fig');
            if ~isempty(fig)
                close(figure(fig));
            end;
            fig = findobj('Tag','3dvrlc2fig');
            if ~isempty(fig)
                close(figure(fig));
            end;
            fig = findobj('Tag','3dvrlc3fig');
            if ~isempty(fig)
                close(figure(fig));
            end;
            fig = findobj('Tag','2dvrlpointloadfig');
            if ~isempty(fig)
                close(figure(fig));
            end;
            fig = findobj('Tag','2dvrlposfig');
            if ~isempty(fig)
                close(figure(fig));
            end;
            fig = findobj('Tag','2dvrlcontfig');
```

```
			if ~isempty(fig)
				close(figure(fig));
			end;
		end;
		fig = findobj('Tag','lfdcGUI');
		saveas(figure(fig),get(fig,'FileName'),'fig')
		close(figure(fig));

	function SetSliders(UserData)
		length=UserData(12);
		years=UserData(15);
		vr=UserData(23);
		loss=UserData(24);
		ccl=UserData(25);
		ati=UserData(26);

		set(findobj(findobj('Tag','lfdcGUI'),'Tag','posslider'),'Max',length);
		set(findobj(findobj('Tag','lfdcGUI'),'Tag','posslider'),'Sliderstep',[1/length 10/length]);
		set(findobj(findobj('Tag','lfdcGUI'),'Tag','posslider'),'Min',1);
		position=round(get(findobj(findobj('Tag','lfdcGUI'),'Tag','posslider'),'Value'));
		if position>length
			set(findobj(findobj('Tag','lfdcGUI'),'Tag','posslider'),'Value',1);
			set(findobj(findobj('Tag','lfdcGUI'),'Tag','Disppos'),'String','1');
		else
			set(findobj(findobj('Tag','lfdcGUI'),'Tag','Disppos'),'String',num2str(position));
		end;

		set(findobj(findobj('Tag','lfdcGUI'),'Tag','yearslider'),'Max',years);
		set(findobj(findobj('Tag','lfdcGUI'),'Tag','yearslider'),'Sliderstep',[1/years 10/years]);
		set(findobj(findobj('Tag','lfdcGUI'),'Tag','yearslider'),'Min',0);
		valueyearslider=round(get(findobj(findobj('Tag','lfdcGUI'),'Tag','yearslider'),'Value'));
		if valueyearslider>years
			set(findobj(findobj('Tag','lfdcGUI'),'Tag','yearslider'),'Value',0);
			set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispyear'),'String','0');
		else
			set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispyear'),'String',num2str(valueyearslider));
		end;

		if vr==1
			set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Max',100);
			set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Sliderstep',[0.01 0.1]);
			set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Min',1);
			valuecontslider=round(get(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value'));
			if valuecontslider>100
				set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value',1);
				set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcont'),'String','1');
			else
				set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcont'),'String',num2str(valuecontslider));
			end;
		elseif loss==1
			set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Max',10e3);
			set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Sliderstep',[0.001 0.01]);
			set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Min',0);
			valuecontslider=round(get(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value'));
			if valuecontslider==0
				set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value',10);
				set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcont'),'String','10');
			else
				set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcont'),'String',num2str(valuecontslider));
			end;
		elseif ccl==1
			set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Max',10e3);
			set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Sliderstep',[0.001 0.01]);
			set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Min',0);
			valuecontslider=round(get(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value'));
			if valuecontslider==0
				set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value',10);
				set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcont'),'String','100000');
			else
				set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcont'),'String',num2str(valuecontslider*100000));
			end;
		elseif ati==1
			set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Max',10);
			set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Sliderstep',[0.11111111 0.5]);
			set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Min',1);
			valuecontslider=round(get(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value'));
			if valuecontslider==0
				set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value',1);
			elseif valuecontslider>10
				set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value',1);
			else
				set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcont'),'String',num2str(valuecontslider));
			end;
		end;

	function [Data,XData,XData123,LR,XD,YData,plotc1cont,plotc2cont,plotc3cont]
						=VRandLoss(UserData,GC,plotc1cont,plotc2cont,plotc3cont)
		rc1=UserData(1);
		rc2=UserData(2);
		rc3=UserData(3);
		costc1=UserData(4);
		costc2=UserData(5);
		costc3=UserData(6);
		ampc1=UserData(7);
		ampc2=UserData(8);
		ampc3=UserData(9);
		sysV=UserData(10);
		Vs=UserData(11);
		length=UserData(12);
		LG=UserData(13);
		IL=UserData(14);
```

```
years=UserData(15);
Interest=UserData(16);
ELF=UserData(17);
stationcost=UserData(18);
stationloss=UserData(19);
w1=UserData(20);
w2=UserData(21);
w4=UserData(22);
vr=UserData(23);
loss=UserData(24);
ccl=UserData(25);
ati=UserData(26);
position=UserData(27);
year=UserData(28);
contour=UserData(29);

Pbase=100e6;                                %Power base - usually chosen as 100 MW
Zbase=(sysV^2)/Pbase;                       %Calculate Zbase from Pbase and SysV

Vspu=Vs/sysV;                               %Calculate the sending voltage in p.u.

XD1=zeros([length,1]);
XD2=zeros([length,1]);
XD3=zeros([length,1]);

testLR1=zeros([(years+1),1]);
testLR2=zeros([(years+1),1]);
testLR3=zeros([(years+1),1]);

for c=1:(years+1)
    XData(c)=(IL/1e6)*(1+LG)^(c-1);                         %Set up XData vector for 3d plot
    DF(c)=[(1+Interest)^(c-1)];                             %Discount Factor
    CLA(c)=8760*(GC(c)/100)*ELF;                            %Cost of losses/kW/Annum [R]
    CL(c)=(IL/1000)*(2*stationloss)*(1+LG)^(c-1);           %Coverter losses [kW]
end;

CCS=2*stationcost*((IL*(1+LG)^(years))/1e6);                %Cost of two converter stations [R]
contourmax1=0;
contourmax2=0;
contourmax3=0;
contourmin1=10;
contourmin2=10;
contourmin3=10;

h = waitbar(0,'Calculating matrices...');

YData=1:length;
for j=1:(years+1)
    LR1(j)=length;
    LR2(j)=length;
    LR3(j)=length;
end;

for i=1:length

    waitbar(i/length)

    R1pu=(rc1*i)/Zbase;                     %Total p.u. resistance for conductor 1
    R2pu=(rc2*i)/Zbase;                     %Total p.u. resistance for conductor 2
    R3pu=(rc3*i)/Zbase;                     %Total p.u. resistance for conductor 3

    XData1(i,:)=XData;
    XData2(i,:)=XData;
    XData3(i,:)=XData;

    for j=1:(years+1)

        Prpu=(IL/Pbase)*(1+LG)^(j-1);                               %Receiving end power in p.u. for year(j-1)
        Vr1pu(i,j)=(Vspu+sqrt((-Vspu)^2-4*0.5*R1pu*Prpu))/2;        %Percentage Voltage Regulation for conductor 1
        Vr2pu(i,j)=(Vspu+sqrt((-Vspu)^2-4*0.5*R2pu*Prpu))/2;        %Percentage Voltage Regulation for conductor 2
        Vr3pu(i,j)=(Vspu+sqrt((-Vspu)^2-4*0.5*R3pu*Prpu))/2;        %Percentage Voltage Regulation for conductor 3
        if isreal(Vr1pu(i,j))
            Vreg1(i,j)=(Vspu-abs(Vr1pu(i,j)))*100;                  %voltage regulation for conductor 1
            Ps1(i,j)=Vspu/(abs(Vr1pu(i,j)))*Prpu*Pbase;             %Sending end power of conductor 1
            Ploss1(i,j)=(Ps1(i,j)-Prpu*Pbase)/1000+CL(j);           %Losses in kW of conductor 1
            ECL1(i,j)=Ploss1(i,j)*CLA(j);                           %Escalated cost of losses [R] of conductor 1
            PVCL1(i,j)=ECL1(i,j)/DF(j);                             %Present Value cost of losses of conductor 1
            PVCL1(:,1)=0;                                           %Present value of year 0 = 0
            jrepeat1=j;
        else
            if testLR1(j)==0
                LR1(j)=i-1;
                testLR1(j)=1;
            end;
            Vreg1(i,j)=Vreg1(i,jrepeat1);
            Ps1(i,j)=Ps1(i,jrepeat1);
            Ploss1(i,j)=Ploss1(i,jrepeat1);
            PVCL1(i,j)=0;
            XData1(i,j)=XData(jrepeat1);
            XD1(i)=1;
        end;
        if isreal(Vr2pu(i,j))
            Vreg2(i,j)=(Vspu-abs(Vr2pu(i,j)))*100;                  %voltage regulation for conductor 2
            Ps2(i,j)=Vspu/(abs(Vr2pu(i,j)))*Prpu*Pbase;             %Sending end power of conductor 2
            Ploss2(i,j)=(Ps2(i,j)-Prpu*Pbase)/1000+CL(j);           %Losses in kW of conductor 2
            ECL2(i,j)=Ploss2(i,j)*CLA(j);                           %Escalated cost of losses [R] of conductor 2
            PVCL2(i,j)=ECL2(i,j)/DF(j);                             %Present Value cost of losses of conductor 2
            PVCL2(:,1)=0;
            jrepeat2=j;
        else
            if testLR2(j)==0
```

```
                LR2(j)=i-1;
                testLR2(j)=1;
            end;
            Vreg2(i,j)=Vreg2(i,jrepeat2);
            XData2(i,j)=XData(jrepeat2);
            Ps2(i,j)=Ps2(i,jrepeat2);
            Ploss2(i,j)=Ploss2(i,jrepeat2);
            PVCL2(i,j)=0;
            XData2(i,j)=XData(jrepeat2);
            XD2(i)=2;
        end;
        if isreal(Vr3pu(i,j))
            Vreg3(i,j)=(Vspu-abs(Vr3pu(i,j)))*100;                    %voltage regulation) for conductor 2
            jrepeat3=j;
            Ps3(i,j)=Vspu/(abs(Vr3pu(i,j)))*Prpu*Pbase;               %Sending end power of conductor 3
            Ploss3(i,j)=(Ps3(i,j)-Prpu*Pbase)/1000+CL(j);             %Losses in kW of conductor 3
            ECL3(i,j)=Ploss3(i,j)*CLA(j);                             %Escalated cost of losses [R] of conductor 3
            PVCL3(i,j)=ECL3(i,j)/DF(j);                               %Present Value cost of losses of conductor 3
            PVCL3(:,1)=0;
        else
            if testLR3(j)==0
                LR3(j)=i-1;
                testLR3(j)=1;
            end;
            Vreg3(i,j)=Vreg3(i,jrepeat3);
            XData3(i,j)=XData(jrepeat3);
            Ps3(i,j)=Ps3(i,jrepeat3);
            Ploss3(i,j)=Ploss3(i,jrepeat3);
            PVCL3(i,j)=0;
            XData3(i,j)=XData(jrepeat3);
            XD3(i)=1;
        end;

        if (vr==1) & (Vreg1(i,j)>=contour)                            %Determine if conductor 1 contour plot is valid
            plotc1cont=1;
        end;

        if (vr==1) & (Vreg2(i,j)>=contour)                            %Determine if conductor 2 contour plot is valid
            plotc2cont=1;
        end;

        if (vr==1) & (Vreg3(i,j)>=contour)                            %Determine if conductor 3 contour plot is valid
            plotc3cont=1;
        end;

        Capacityc1(i,j)=(CCS+costc1*i)/(2*sysV*1e-3*ampc1*1*i);       %Capacity Matrix of conductor 1
        Capacityc2(i,j)=(CCS+costc2*i)/(2*sysV*1e-3*ampc2*1*i);       %Capacity Matrix of conductor 2
        Capacityc3(i,j)=(CCS+costc3*i)/(2*sysV*1e-3*ampc3*1*i);       %Capacity Matrix of conductor 3

    end;

    CPVCL1(i,:)=cumsum(PVCL1(i,:));                                   %Cumulative Present Value Cost of Losses for Conductor 1
    CPVCL2(i,:)=cumsum(PVCL2(i,:));                                   %Cumulative Present Value Cost of Losses for Conductor 2
    CPVCL3(i,:)=cumsum(PVCL3(i,:));                                   %Cumulative Present Value Cost of Losses for Conductor 3

    PVCCL1(i,:)=CPVCL1(i,:)+costc1*i;                                 %Cumulative Present Value of Capital and Cost of Losses
    PVCCL2(i,:)=CPVCL2(i,:)+costc2*i;                                 %Cumulative Present Value of Capital and Cost of Losses
    PVCCL3(i,:)=CPVCL3(i,:)+costc3*i;                                 %Cumulative Present Value of Capital and Cost of Losses

    PVCCL1(i,:)=(PVCCL1(i,:)+CCS)/1e6;                                %Add cost of two converter stations
    PVCCL2(i,:)=(PVCCL2(i,:)+CCS)/1e6;
    PVCCL3(i,:)=(PVCCL3(i,:)+CCS)/1e6;

    for n=1:(years+1)
        if (ccl==1) & (PVCCL1(i,n)>=contour*10000) & (contour*10000>PVCCL1(1,1))
            plotc1cont=1;
        elseif (loss==1) & (Ploss1(i,n)>=contour) & (contour>Ploss1(1,1))
            plotc1cont=1;
        end;
        if (ccl==1) & (PVCCL2(i,n)>=contour*10000) & (contour*10000>PVCCL2(1,1))
            plotc2cont=1;
        elseif (loss==1) & (Ploss2(i,n)>=contour) & (contour>Ploss2(1,1))
            plotc2cont=1;
        end;
        if (ccl==1) & (PVCCL3(i,n)>=contour*10000) & (contour*10000>PVCCL3(1,1))
            plotc3cont=1;
        elseif (loss==1) & (Ploss3(i,n)>=contour) & (contour>Ploss3(1,1))
            plotc3cont=1;
        end;
    end;
end;

close(h);

if isreal(Vr1pu)
    set(findobj(findobj('Tag','lfdcGUI'),'Tag','c1t'),'BackgroundColor',[.8314 .8157 .7843]);
else
    set(findobj(findobj('Tag','lfdcGUI'),'Tag','c1t'),'BackgroundColor','r');
end;
if isreal(Vr2pu)
    set(findobj(findobj('Tag','lfdcGUI'),'Tag','c2t'),'BackgroundColor',[.8314 .8157 .7843]);
else
    set(findobj(findobj('Tag','lfdcGUI'),'Tag','c2t'),'BackgroundColor','r');
end;
if isreal(Vr3pu)
    set(findobj(findobj('Tag','lfdcGUI'),'Tag','c3t'),'BackgroundColor',[.8314 .8157 .7843]);
else
    set(findobj(findobj('Tag','lfdcGUI'),'Tag','c3t'),'BackgroundColor','r');
end;

%Calculate upper, median and lower values for LCC KPI
```

```
LCClower=0;
LCCupper=CCS;
if PVCCL1(length,years+1)>PVCCL2(length,years+1) & PVCCL1(length,years+1)>PVCCL3(length,years+1)
        LCClower=PVCCL1(length,years+1)*1e6+1e6;
elseif PVCCL2(length,years+1)>PVCCL3(length,years+1)
        LCClower=PVCCL2(length,years+1)*1e6+1e6;
else
        LCClower=PVCCL3(length,years+1)*1e6+1e6;
end;
LCCmedian=LCClower-(((LCClower-LCCupper)/10)*3);

%Calculate upper, median and lower values for Capacity KPI
Capacitylower=0;
Capacityupper=0;
if Capacityc1(length,1)<Capacityc2(length,1) & Capacityc1(length,1)<Capacityc3(length,1)
        Capacityupper=Capacityc1(length,1)-1;
elseif Capacityc2(length,1)<Capacityc3(length,1)
        Capacityupper=Capacityc2(length,1)-1;
else
        Capacityupper=Capacityc3(length,1)-1;
end;
if Capacityc1(1,1)>Capacityc2(1,1) & Capacityc1(1,1)>Capacityc3(1,1)
        Capacitylower=Capacityc1(1,1)+1;
elseif Capacityc2(1,1)>Capacityc3(1,1)
        Capacitylower=Capacityc2(1,1)+1;
else;
        Capacitylower=Capacityc3(1,1)+1;
end;
Capacitymedian=Capacitylower-(((Capacitylower-Capacityupper)/10)*3);

%Calculate upper, median and lower values for Voltage Regulation KPI
VRupper=0.000001;
VRlower=0.5;
VRmedian=VRlower-(((VRlower-VRupper)/10)*3);

h = waitbar(0,'Calculating ATIs...');

for i=1:length
        waitbar(i/length)
        for j=1:years+1
                LCCc1(i,j)=10-(PVCCL1(i,j)*1e6-LCCupper)/((LCClower-LCCupper)/10);
                LCCc2(i,j)=10-(PVCCL2(i,j)*1e6-LCCupper)/((LCClower-LCCupper)/10);
                LCCc3(i,j)=10-(PVCCL3(i,j)*1e6-LCCupper)/((LCClower-LCCupper)/10);

                Capacityc1(i,j)=10-(Capacityc1(i,j)-Capacityupper)/((Capacitylower-Capacityupper)/10);
                Capacityc2(i,j)=10-(Capacityc2(i,j)-Capacityupper)/((Capacitylower-Capacityupper)/10);
                Capacityc3(i,j)=10-(Capacityc3(i,j)-Capacityupper)/((Capacitylower-Capacityupper)/10);

                VRc1(i,j)=10-((Vreg1(i,j)/100)-VRupper)/((VRlower-VRupper)/10);
                VRc2(i,j)=10-((Vreg2(i,j)/100)-VRupper)/((VRlower-VRupper)/10);
                VRc3(i,j)=10-((Vreg3(i,j)/100)-VRupper)/((VRlower-VRupper)/10);

                ATI1(i,j)=LCCc1(i,j)*w1+Capacityc1(i,j)*w2+VRc1(i,j)*w4;
                ATI2(i,j)=LCCc2(i,j)*w1+Capacityc2(i,j)*w2+VRc2(i,j)*w4;
                ATI3(i,j)=LCCc3(i,j)*w1+Capacityc3(i,j)*w2+VRc3(i,j)*w4;

                %Determine minimum and maximum values for ATI contour plot
                if (ati==1) & (ATI1(i,j)>contourmax1)
                        contourmax1=ATI1(i,j);
                end;
                if (ati==1) & (ATI2(i,j)>contourmax2)
                        contourmax2=ATI2(i,j);
                end;
                if (ati==1) & (ATI3(i,j)>contourmax3)
                        contourmax3=ATI3(i,j);
                end;

                if (ati==1) & (ATI1(i,j)<contourmin1)
                        contourmin1=ATI1(i,j);
                end;
                if (ati==1) & (ATI2(i,j)<contourmin2)
                        contourmin2=ATI2(i,j);
                end;
                if (ati==1) & (ATI3(i,j)<contourmin3)
                        contourmin3=ATI3(i,j);
                end;

        end;
end;

close(h);
%Determine if contour plot is valid for ATIs of conductor 1,2 and 3
if (ati==1) & (contour>=contourmin1) & (contour<=contourmax1)
        plotc1cont=1;
end;
if (ati==1) & (contour>=contourmin2) & (contour<=contourmax2)
        plotc2cont=1;
end;
if (ati==1) & (contour>=contourmin3) & (contour<=contourmax3)
        plotc3cont=1;
end;

Data(:,:,1) = Vreg1;
Data(:,:,2) = Vreg2;
Data(:,:,3) = Vreg3;
Data(:,:,4) = Ploss1;
Data(:,:,5) = Ploss2;
Data(:,:,6) = Ploss3;
Data(:,:,7) = PVCCL1;
Data(:,:,8) = PVCCL2;
Data(:,:,9) = PVCCL3;
```

```
Data(:,:,10) = ATI1;
Data(:,:,11) = ATI2;
Data(:,:,12) = ATI3;

XData123(:,:,1) = XData1;
XData123(:,:,2) = XData2;
XData123(:,:,3) = XData3;

XD(:,:,1) =     XD1;
XD(:,:,2) =     XD2;
XD(:,:,3) =     XD3;

LR(:,:,1) =     LR1;
LR(:,:,2) =     LR2;
LR(:,:,3) =     LR3;
```

B.4.3 Plot2d.m

```
function plot2d(XData,XData123,LR,XD,YData,namec1,namec2,namec3,plotc1cont,plotc2cont,plotc3cont,UserData,Data)

	vr		=	UserData(23);
	loss		=	UserData(24);
	ccl		=	UserData(25);
	ati		=	UserData(26);
	position	=	UserData(27);
	year		=	UserData(28);
	contour		=	UserData(29);

	Vreg1		=	Data(:,:,1);
	Vreg2		=	Data(:,:,2);
	Vreg3		=	Data(:,:,3);
	Ploss1		=	Data(:,:,4);
	Ploss2		=	Data(:,:,5);
	Ploss3		=	Data(:,:,6);
	PVCCL1		=	Data(:,:,7);
	PVCCL2		=	Data(:,:,8);
	PVCCL3		=	Data(:,:,9);
	ATI1		=	Data(:,:,10);
	ATI2		=	Data(:,:,11);
	ATI3		=	Data(:,:,12);

	XData1		=	XData123(:,:,1);
	XData2		=	XData123(:,:,2);
	XData3		=	XData123(:,:,3);

	XD1		=	XD(:,:,1);
	XD2		=	XD(:,:,2);
	XD3		=	XD(:,:,3);

	LR1		=	LR(:,:,1);
	LR2		=	LR(:,:,2);
	LR3		=	LR(:,:,3);

	Vregpos1	=	Vreg1(position,:);
	Vregpos2	=	Vreg2(position,:);
	Vregpos3	=	Vreg3(position,:);
	losspos1	=	Ploss1(position,:);
	losspos2	=	Ploss2(position,:);
	losspos3	=	Ploss3(position,:);
	PVCCLpos1	=	PVCCL1(position,:);
	PVCCLpos2	=	PVCCL2(position,:);
	PVCCLpos3	=	PVCCL3(position,:);
	ATIpos1		=	ATI1(position,:);
	ATIpos2		=	ATI2(position,:);
	ATIpos3		=	ATI3(position,:);

	XDatapos1	=	XData1(position,:);
	XDatapos2	=	XData2(position,:);
	XDatapos3	=	XData3(position,:);

	YData1		=	1:LR1(year+1);
	YData2		=	1:LR2(year+1);
	YData3		=	1:LR3(year+1);

	for i1=1:LR1(year+1);
		Vreg1LR(i1)=Vreg1(i1,year+1);
		loss1LR(i1)=Ploss1(i1,year+1);
		PVCCL1LR(i1)=PVCCL1(i1,year+1);
		ATI1LR(i1)=ATI1(i1,year+1);
	end;
	for i2=1:LR2(year+1);
		Vreg2LR(i2)=Vreg2(i2,year+1);
		loss2LR(i2)=Ploss2(i2,year+1);
		PVCCL2LR(i2)=PVCCL2(i2,year+1);
		ATI2LR(i2)=ATI2(i2,year+1);
	end;
	for i3=1:LR3(year+1);
		Vreg3LR(i3)=Vreg3(i3,year+1);
		loss3LR(i3)=Ploss3(i3,year+1);
		PVCCL3LR(i3)=PVCCL3(i3,year+1);
		ATI3LR(i3)=ATI3(i3,year+1);
	end;

	axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrlposaxes'))
	cla;
	if vr==1
		if XD1(position)==0;
			H=plot(XData,Vregpos1,'r');
			hold on;
			set(gca,'Tag','vrlposaxes');
			set(H,'LineWidth',1);
		else
			H=plot(XDatapos1,Vregpos1,'r');
			hold on;
			set(gca,'Tag','vrlposaxes');
			set(H,'LineWidth',1);
		end;
		if XD2(position)==0;
			H=plot(XData,Vregpos2,'b');
			hold on;
			set(gca,'Tag','vrlposaxes');
			set(H,'LineWidth',1);
		else
			H=plot(XDatapos2,Vregpos2,'b');
```

```
                hold on;
                set(gca,'Tag','vrlposaxes');
                set(H,'LineWidth',1);
            end;
            if XD3(position)==0;
                H=plot(XData,Vregpos3,'g');
                hold off;
                set(gca,'Tag','vrlposaxes');
                set(H,'LineWidth',1);
            else
                H=plot(XDatapos3,Vregpos3,'g');
                hold off;
                set(gca,'Tag','vrlposaxes');
                set(H,'LineWidth',1);
            end;
            ylabel('Voltage Regulation [%]');
            title('Voltage Regulation vs Point Load');
        elseif loss==1
            if XD1(position)==0;
                H=plot(XData,losspos1,'r');
                hold on;
                set(gca,'Tag','vrlposaxes');
                set(H,'LineWidth',1);
            else
                H=plot(XDatapos1,losspos1,'r');
                hold on;
                set(gca,'Tag','vrlposaxes');
                set(H,'LineWidth',1);
            end;
            if XD2(position)==0;
                H=plot(XData,losspos2,'b');
                hold on;
                set(gca,'Tag','vrlposaxes');
                set(H,'LineWidth',1);
            else
                H=plot(XDatapos2,losspos2,'b');
                hold on;
                set(gca,'Tag','vrlposaxes');
                set(H,'LineWidth',1);
            end;
            if XD3(position)==0;
                H=plot(XData,losspos3,'g');
                hold off;
                set(gca,'Tag','vrlposaxes');
                set(H,'LineWidth',1);
            else
                H=plot(XDatapos3,losspos3,'g');
                hold off;
                set(gca,'Tag','vrlposaxes');
                set(H,'LineWidth',1);
            end;
            ylabel('Power Losses [kW]');
            title('Power Losses vs Point Load');
        elseif ccl==1
            if XD1(position)==0;
                H=plot(XData,PVCCLpos1,'r');
                hold on;
                set(gca,'Tag','vrlposaxes');
                set(H,'LineWidth',1);
            else
                H=plot(XDatapos1,PVCCLpos1,'r');
                hold on;
                set(gca,'Tag','vrlposaxes');
                set(H,'LineWidth',1);
            end;
            if XD2(position)==0;
                H=plot(XData,PVCCLpos2,'b');
                hold on;
                set(gca,'Tag','vrlposaxes');
                set(H,'LineWidth',1);
            else
                H=plot(XDatapos2,PVCCLpos2,'b');
                hold on;
                set(gca,'Tag','vrlposaxes');
                set(H,'LineWidth',1);
            end;
            if XD3(position)==0;
                H=plot(XData,PVCCLpos3,'g');
                hold off;
                set(gca,'Tag','vrlposaxes');
                set(H,'LineWidth',1);
            else
                H=plot(XDatapos3,PVCCLpos3,'g');
                hold off;
                set(gca,'Tag','vrlposaxes');
                set(H,'LineWidth',1);
            end;
            ylabel('Cumulative Cost [MR]');
            title('LCC vs Point Load');
        elseif ati==1
            if XD1(position)==0;
                H=plot(XData,ATIpos1,'r');
                hold on;
                set(gca,'Tag','vrlposaxes');
                set(H,'LineWidth',1);
            else
                H=plot(XDatapos1,ATIpos1,'r');
                hold on;
                set(gca,'Tag','vrlposaxes');
                set(H,'LineWidth',1);
            end;
```

```
    if XD2(position)==0;
        H=plot(XData,ATIpos2,'b');
        hold on;
        set(gca,'Tag','vrlposaxes');
        set(H,'LineWidth',1);
    else
        H=plot(XDatapos2,ATIpos2,'b');
        hold on;
        set(gca,'Tag','vrlposaxes');
        set(H,'LineWidth',1);
    end;
    if XD3(position)==0;
        H=plot(XData,ATIpos3,'g');
        hold off;
        set(gca,'Tag','vrlposaxes');
        set(H,'LineWidth',1);
    else
        H=plot(XDatapos3,ATIpos3,'g');
        hold off;
        set(gca,'Tag','vrlposaxes');
        set(H,'LineWidth',1);
    end;
    ylabel('ATI');
    title('ATI vs Point Load');
end;
set(gca,'Tag','vrlposaxes');
axis tight
a = axis;
AXIS([a(1)-(a(2)-a(1))*0.1 a(2)+(a(2)-a(1))*0.1 a(3)-(a(4)-a(3))*0.1 a(4)+(a(4)-a(3))*0.1])
xlabel('Point Load [MW]');
legend(namec1,namec2,namec3,0);
grid on;

if get(findobj(findobj('Tag','lfdcGUI'),'Tag','hrfcb'),'Value')
    fsize = 36;
    fig = findobj('Tag','2dvrlpointloadfig');
    if isempty(fig)
        figure
        if vr==1
            set(gcf,'IntegerHandle','off',...
            'Name','Voltage Regulation vs Point Load',...
            'NumberTitle','off',...
            'Tag','2dvrlpointloadfig');
        elseif loss==1
            set(gcf,'IntegerHandle','off',...
            'Name','Power Losses vs Point Load',...
            'NumberTitle','off',...
            'Tag','2dvrlpointloadfig');
        elseif ccl==1
            set(gcf,'IntegerHandle','off',...
            'Name','LCC vs Point Load',...
            'NumberTitle','off',...
            'Tag','2dvrlpointloadfig');
        elseif ati==1
            set(gcf,'IntegerHandle','off',...
            'Name','ATI vs Point Load',...
            'NumberTitle','off',...
            'Tag','2dvrlpointloadfig');
        end;
    else
        figure(fig)
        if vr==1
            set(gcf,'IntegerHandle','off',...
            'Name','Voltage Regulation vs Point Load',...
            'NumberTitle','off',...
            'Tag','2dvrlpointloadfig');
        elseif loss==1
            set(gcf,'IntegerHandle','off',...
            'Name','Power Losses vs Point Load',...
            'NumberTitle','off',...
            'Tag','2dvrlpointloadfig');
        elseif ccl==1
            set(gcf,'IntegerHandle','off',...
            'Name','LCC vs Point Load',...
            'NumberTitle','off',...
            'Tag','2dvrlpointloadfig');
        elseif ati==1
            set(gcf,'IntegerHandle','off',...
            'Name','ATI vs Point Load',...
            'NumberTitle','off',...
            'Tag','2dvrlpointloadfig');
        end;
    end;
    set(gcf,'Position',[1 29 1024 672]);
    %set(gca,'Position',[0.15 0.15 0.7 0.7]);
    set(gca,'Position',[0.15 0.15 0.82 0.82]);
    if vr==1
        if XD1(position)==0;
            H=plot(XData,Vregpos1,'r');
            hold on;
            set(H,'LineWidth',1.5);
        else
            H=plot(XDatapos1,Vregpos1,'r');
            hold on;
            set(H,'LineWidth',1.5);
        end;
        if XD2(position)==0;
            H=plot(XData,Vregpos2,'b');
            hold on;
            set(H,'LineWidth',1.5);
        else
```

```
				H=plot(XDatapos2,Vregpos2,'b');
				hold on;
				set(H,'LineWidth',1.5);
			end;
			if XD3(position)==0;
				H=plot(XData,Vregpos3,'g');
				hold off;
				set(H,'LineWidth',1.5);
			else
				H=plot(XDatapos3,Vregpos3,'g');
				hold off;
				set(H,'LineWidth',1.5);
			end;
			hy=ylabel('Voltage Regulation [%]');
			ht=title('Voltage Regulation vs Point Load');
		elseif loss==1
			if XD1(position)==0;
				H=plot(XData,losspos1,'r');
				hold on;
				set(H,'LineWidth',1.5);
			else
				H=plot(XDatapos1,losspos1,'r');
				hold on;
				set(H,'LineWidth',1.5);
			end;
			if XD2(position)==0;
				H=plot(XData,losspos2,'b');
				hold on;
				set(H,'LineWidth',1.5);
			else
				H=plot(XDatapos2,losspos2,'b');
				hold on;
				set(H,'LineWidth',1.5);
			end;
			if XD3(position)==0;
				H=plot(XData,losspos3,'g');
				hold off;
				set(H,'LineWidth',1.5);
			else
				H=plot(XDatapos3,losspos3,'g');
				hold off;
				set(H,'LineWidth',1.5);
			end;
			hy=ylabel('Power Losses [kW]');
			ht=title('Power Losses vs Point Load');
		elseif ccl==1
			if XD1(position)==0;
				H=plot(XData,PVCCLpos1,'r');
				hold on;
				set(H,'LineWidth',1.5);
			else
				H=plot(XDatapos1,PVCCLpos1,'r');
				hold on;
				set(H,'LineWidth',1.5);
			end;
			if XD2(position)==0;
				H=plot(XData,PVCCLpos2,'b');
				hold on;
				set(H,'LineWidth',1.5);
			else
				H=plot(XDatapos2,PVCCLpos2,'b');
				hold on;
				set(H,'LineWidth',1.5);
			end;
			if XD3(position)==0;
				H=plot(XData,PVCCLpos3,'g');
				hold off;
				set(H,'LineWidth',1.5);
			else
				H=plot(XDatapos3,PVCCLpos3,'g');
				hold off;
				set(H,'LineWidth',1.5);
			end;
			hy=ylabel('Cumulative Cost [MR]');
			ht=title('LCC vs Point Load');
		elseif ati==1
			if XD1(position)==0;
				H=plot(XData,ATIpos1,'r');
				hold on;
				set(H,'LineWidth',1.5);
			else
				H=plot(XDatapos1,ATIpos1,'r');
				hold on;
				set(H,'LineWidth',1.5);
			end;
			if XD2(position)==0;
				H=plot(XData,ATIpos2,'b');
				hold on;
				set(H,'LineWidth',1.5);
			else
				H=plot(XDatapos2,ATIpos2,'b');
				hold on;
				set(H,'LineWidth',1.5);
			end;
			if XD3(position)==0;
				H=plot(XData,ATIpos3,'g');
				hold off;
				set(H,'LineWidth',1.5);
			else
				H=plot(XDatapos3,ATIpos3,'g');
				hold off;
```

```
                set(H,'LineWidth',1.5);
            end;
            hy=ylabel('ATI');
            ht=title('ATI vs Point Load');
        end;
        hx=xlabel('Point Load [MW]');
        set(hx,'FontSize',fsize*0.8,
        'VerticalAlignment','top');
        set(hy,'FontSize',fsize*0.8,
        'VerticalAlignment','bottom');
        set(ht,'FontSize',fsize);
        set(gcf,'Color',[1 1 1]);
        set(gca,'FontSize',fsize*0.7);
        axis tight
        a = axis;
        AXIS([a(1)-(a(2)-a(1))*0.1 a(2)+(a(2)-a(1))*0.1 a(3)-(a(4)-a(3))*0.1 a(4)+(a(4)-a(3))*0.1])
        legend(namec1,namec2,namec3,0);
        grid on;
end;

%Voltage Regulation/Power Losses/LCC vs Position%
axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrlyearaxes'))
cla;
if vr==1
        H=plot(YData1,Vreg1LR,'r');
        hold on;
        set(gca,'Tag','vrlyearaxes');
        set(H,'LineWidth',1)
        H=plot(YData2,Vreg2LR,'b');
        hold on;
        set(gca,'Tag','vrlyearaxes');
        set(H,'LineWidth',1)
        H=plot(YData3,Vreg3LR,'g');
        hold off;
        set(H,'LineWidth',1)
        ylabel('Voltage Regulation [%]');
        title('Voltage Regulation vs Position');
elseif loss==1
        H=plot(YData1,loss1LR,'r');
        hold on;
        set(gca,'Tag','vrlyearaxes');
        set(H,'LineWidth',1)
        H=plot(YData2,loss2LR,'b');
        hold on;
        set(gca,'Tag','vrlyearaxes');
        set(H,'LineWidth',1)
        H=plot(YData3,loss3LR,'g');
        hold off;
        set(H,'LineWidth',1)
        ylabel('Power Losses [kW]');
        title('Power Losses vs Position');
elseif ccl==1
        H=plot(YData1,PVCCL1LR,'r');
        hold on;
        set(gca,'Tag','vrlyearaxes');
        set(H,'LineWidth',1)
        H=plot(YData2,PVCCL2LR,'b');
        hold on;
        set(gca,'Tag','vrlyearaxes');
        set(H,'LineWidth',1)
        H=plot(YData3,PVCCL3LR,'g');
        hold off;
        set(H,'LineWidth',1)
        ylabel('Cumulative Cost [MR]');
        title('LCC vs Position');
elseif ati==1
        H=plot(YData1,ATI1LR,'r');
        hold on;
        set(gca,'Tag','vrlyearaxes');
        set(H,'LineWidth',1)
        H=plot(YData2,ATI2LR,'b');
        hold on;
        set(gca,'Tag','vrlyearaxes');
        set(H,'LineWidth',1)
        H=plot(YData3,ATI3LR,'g');
        hold off;
        set(H,'LineWidth',1)
        ylabel('ATI');
        title('ATI vs Position');
end;
set(gca,'Tag','vrlyearaxes');
axis tight
a = axis;
AXIS([a(1)-(a(2)-a(1))*0.1 a(2)+(a(2)-a(1))*0.1 a(3)-(a(4)-a(3))*0.1 a(4)+(a(4)-a(3))*0.1])
xlabel('Distance');
%legend(namec1,namec2,namec3,0);
grid on;

if get(findobj(findobj('Tag','lfdcGUI'),'Tag','hrfcb'),'Value')
        fsize = 36;
        fig = findobj('Tag','2dvrlposfig');
        if isempty(fig)
            figure
            if vr==1
                set(gcf,'IntegerHandle','off',...
                'Name','Voltage Regulation vs Position',...
                'NumberTitle','off',...
                'Tag','2dvrlposfig');
            elseif loss==1
                set(gcf,'IntegerHandle','off',...
                'Name','Power Losses vs Position',...
```

```
                'NumberTitle','off',...
                'Tag','2dvrlposfig');
        elseif ccl==1
                set(gcf,'IntegerHandle','off',...
                'Name','LCC vs Position',...
                'NumberTitle','off',...
                'Tag','2dvrlposfig');
        elseif ati==1
                set(gcf,'IntegerHandle','off',...
                'Name','ATI vs Position',...
                'NumberTitle','off',...
                'Tag','2dvrlposfig');
        end;
else
        figure(fig)
        if vr==1
                set(gcf,'IntegerHandle','off',...
                'Name','Voltage Regulation vs Position',...
                'NumberTitle','off',...
                'Tag','2dvrlposfig');
        elseif loss==1
                set(gcf,'IntegerHandle','off',...
                'Name','Power Losses vs Position',...
                'NumberTitle','off',...
                'Tag','2dvrlposfig');
        elseif ccl==1
                set(gcf,'IntegerHandle','off',...
                'Name','LCC vs Position',...
                'NumberTitle','off',...
                'Tag','2dvrlposfig');
        elseif ati==1
                set(gcf,'IntegerHandle','off',...
                'Name','ATI vs Position',...
                'NumberTitle','off',...
                'Tag','2dvrlposfig');
        end;
end;
set(gcf,'Position',[1 29 1024 672]);
set(gca,'Position',[0.15 0.15 0.82 0.82]);
if vr==1
        H=plot(YData1,Vreg1LR,'r');
        hold on;
        set(H,'LineWidth',1.5);
        H=plot(YData2,Vreg2LR,'b');
        hold on;
        set(H,'LineWidth',1.5);
        H=plot(YData3,Vreg3LR,'g');
        hold off;
        set(H,'LineWidth',1.5);
        hy=ylabel('Voltage Regulation [%]');
        ht=title('Voltage Regulation vs Position');
elseif loss==1
        H=plot(YData1,loss1LR,'r');
        hold on;
        set(H,'LineWidth',1.5);
        H=plot(YData2,loss2LR,'b');
        hold on;
        set(H,'LineWidth',1.5);
        H=plot(YData3,loss3LR,'g');
        hold off;
        set(H,'LineWidth',1.5);
        hy=ylabel('Power Losses [kW]');
        ht=title('Power Losses vs Position');
elseif ccl==1
        H=plot(YData1,PVCCL1LR,'r');
        hold on;
        set(H,'LineWidth',1.5);
        H=plot(YData2,PVCCL2LR,'b');
        hold on;
        set(H,'LineWidth',1.5);
        H=plot(YData3,PVCCL3LR,'g');
        hold off;
        set(H,'LineWidth',1.5);
        hy=ylabel('Cumulative Cost [MR]');
        ht=title('LCC vs Position');
elseif ati==1
        H=plot(YData1,ATI1LR,'r');
        hold on;
        set(H,'LineWidth',1.5);
        H=plot(YData2,ATI2LR,'b');
        hold on;
        set(H,'LineWidth',1.5);
        H=plot(YData3,ATI3LR,'g');
        hold off;
        set(H,'LineWidth',1.5);
        hy=ylabel('ATI');
        ht=title('ATI vs Position');
end;

hx=xlabel('Distance [km]');
set(hx,'FontSize',fsize*0.8,...
'VerticalAlignment','top');
set(hy,'FontSize',fsize*0.8,...
'VerticalAlignment','bottom');
set(ht,'FontSize',fsize);
set(gcf,'Color',[1 1 1]);
set(gca,'FontSize',fsize*0.7);
axis tight
a = axis;
AXIS([a(1)-(a(2)-a(1))*0.1 a(2)+(a(2)-a(1))*0.1 a(3)-(a(4)-a(3))*0.1 a(4)+(a(4)-a(3))*0.1])
legend(namec1,namec2,namec3,0);
```

```
        grid on;
    end;

    %Contour plots of Voltage regulation/losses/Cumulative Cost
    if ~isempty(plotc1cont) & ~isempty(plotc2cont) & ~isempty(plotc3cont)
        axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrlcontaxes'))
        cla;
        if vr==1
            [cs1,h1] = contour(XData,YData,Vreg1,[contour contour],'r');
            hold on;
            set(gca,'Tag','vrlcontaxes');
            [cs2,h2] = contour(XData,YData,Vreg2,[contour contour],'b');
            hold on;
            set(gca,'Tag','vrlcontaxes');
            [cs3,h3] = contour(XData,YData,Vreg3,[contour contour],'g');
            hold off;
            title('Contour plot of Voltage Regulation');
        elseif loss==1
            [cs1,h1] = contour(XData,YData,Ploss1,[contour contour],'r');
            hold on;
            set(gca,'Tag','vrlcontaxes');
            [cs2,h2] = contour(XData,YData,Ploss2,[contour contour],'b');
            hold on;
            set(gca,'Tag','vrlcontaxes');
            [cs3,h3] = contour(XData,YData,Ploss3,[contour contour],'g');
            hold off;
            title('Contour plot of Power Losses');
        elseif ccl==1
            [cs1,h1] = contour(XData,YData,PVCCL1,[contour*10000 contour*10000],'r');
            hold on;
            set(gca,'Tag','vrlcontaxes');
            [cs2,h2] = contour(XData,YData,PVCCL2,[contour*10000 contour*10000],'b');
            hold on;
            set(gca,'Tag','vrlcontaxes');
            [cs3,h3] = contour(XData,YData,PVCCL3,[contour*10000 contour*10000],'g');
            hold off;
            title('Contour plot of LCC [MR]');
        elseif ati==1
            [cs1,h1] = contour(XData,YData,ATI1,[contour contour],'r');
            hold on;
            set(gca,'Tag','vrlcontaxes');
            [cs2,h2] = contour(XData,YData,ATI2,[contour contour],'b');
            hold on;
            set(gca,'Tag','vrlcontaxes');
            [cs3,h3] = contour(XData,YData,ATI3,[contour contour],'g');
            hold off;
            title('Contour plot of ATI');
        end;
        set(gca,'Tag','vrlcontaxes');
        clabel(cs1,h1)%,'labelspacing',100)
        clabel(cs2,h2)%,'labelspacing',100)
        clabel(cs3,h3)%,'labelspacing',100)
        ylabel('Distance [km]');
        xlabel('Point Load [MW]');
        legend(namec1,namec2,namec3,0);

        if get(findobj(findobj('Tag','lfdcGUI'),'Tag','hrfcb'),'Value')
            fsize = 36;
            fig = findobj('Tag','2dvrlcontfig');
            if isempty(fig)
                figure
                if vr==1
                    set(gcf,'IntegerHandle','off',...
                    'Name','Voltage Regulation Contour Plot',...
                    'NumberTitle','off',...
                    'Tag','2dvrlcontfig');
                elseif loss==1
                    set(gcf,'IntegerHandle','off',...
                    'Name','Power Losses Contour Plot',...
                    'NumberTitle','off',...
                    'Tag','2dvrlcontfig');
                elseif ccl==1
                    set(gcf,'IntegerHandle','off',...
                    'Name','Cumulative Cost Contour Plot',...
                    'NumberTitle','off',...
                    'Tag','2dvrlcontfig');
                elseif ati==1
                    set(gcf,'IntegerHandle','off',...
                    'Name','ATI Contour Plot',...
                    'NumberTitle','off',...
                    'Tag','2dvrlcontfig');
                end;
            else
                figure(fig)
                if vr==1
                    set(gcf,'IntegerHandle','off',...
                    'Name','Voltage Regulation Contour Plot',...
                    'NumberTitle','off',...
                    'Tag','2dvrlcontfig');
                elseif loss==1
                    set(gcf,'IntegerHandle','off',...
                    'Name','Power Losses Contour Plot',...
                    'NumberTitle','off',...
                    'Tag','2dvrlcontfig');
                elseif ccl==1
                    set(gcf,'IntegerHandle','off',...
                    'Name','Cumulative Cost Contour Plot',...
                    'NumberTitle','off',...
                    'Tag','2dvrlcontfig');
                elseif ati==1
                    set(gcf,'IntegerHandle','off',...
```

```
                'Name','ATI Contour Plot',...
                'NumberTitle','off',...
                'Tag','2dvrlcontfig');
            end;
        end;
        set(gcf,'Position',[1 29 1024 672]);
        set(gca,'Position',[0.15 0.15 0.82 0.82]);
        if vr==1
            [cs1,h] = contour(XData,YData,Vreg1,[contour contour],'r');
            set(h,'LineWidth',1.5);
            clabel(cs1,h);
            hold on;
            [cs2,h] = contour(XData,YData,Vreg2,[contour contour],'b');
            set(h,'LineWidth',1.5);
            clabel(cs2,h);
            hold on;
            [cs3,h] = contour(XData,YData,Vreg3,[contour contour],'g');
            set(h,'LineWidth',1.5);
            clabel(cs3,h);
            hold off;
            ht=title('Contour plot of Voltage Regulation');
        elseif loss==1
            [cs1,h] = contour(XData,YData,Ploss1,[contour contour],'r');
            set(h,'LineWidth',1.5);
            clabel(cs1,h);
            hold on;
            [cs2,h] = contour(XData,YData,Ploss2,[contour contour],'b');
            set(h,'LineWidth',1.5);
            clabel(cs2,h);
            hold on;
            [cs3,h] = contour(XData,YData,Ploss3,[contour contour],'g');
            set(h,'LineWidth',1.5);
            clabel(cs3,h);
            hold off;
            ht=title('Contour plot of Power Losses');
        elseif ccl==1
            [cs1,h] = contour(XData,YData,PVCCL1,[contour*10000 contour*10000],'r');
            set(h,'LineWidth',1.5);
            clabel(cs1,h);
            hold on;
            [cs2,h] = contour(XData,YData,PVCCL2,[contour*10000 contour*10000],'b');
            set(h,'LineWidth',1.5);
            clabel(cs2,h);
            hold on;
            [cs3,h] = contour(XData,YData,PVCCL3,[contour*10000 contour*10000],'g');
            set(h,'LineWidth',1.5);
            clabel(cs3,h);
            hold off;
            ht=title('Contour plot of Cumulative Cost');
        elseif ati==1
            [cs1,h] = contour(XData,YData,ATI1,[contour contour],'r');
            set(h,'LineWidth',1.5);
            clabel(cs1,h);
            hold on;
            [cs2,h] = contour(XData,YData,ATI2,[contour contour],'b');
            set(h,'LineWidth',1.5);
            clabel(cs2,h);
            hold on;
            [cs3,h] = contour(XData,YData,ATI3,[contour contour],'g');
            set(h,'LineWidth',1.5);
            clabel(cs3,h);
            hold off;
            ht=title('Contour plot of ATI');
        end;
        hy=ylabel('Distance [km]');
        hx=xlabel('Point Load [MW]');
        set(hx,'FontSize',fsize*0.8,...
        'VerticalAlignment','top');
        set(hy,'FontSize',fsize*0.8,...
        'VerticalAlignment','bottom');
        set(ht,'FontSize',fsize);
        set(gcf,'Color',[1 1 1]);
        set(gca,'FontSize',fsize*0.7);
        axis tight
        legend(namec1,namec2,namec3,0);
    end;

elseif ~isempty(plotc1cont) & ~isempty(plotc2cont) & isempty(plotc3cont)
    axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrlcontaxes'))
    cla;
    if vr==1
        [cs1,h1] = contour(XData,YData,Vreg1,[contour contour],'r');
        hold on;
        set(gca,'Tag','vrlcontaxes');
        [cs2,h2] = contour(XData,YData,Vreg2,[contour contour],'b');
        hold off;
        title('Contour plot of Voltage Regulation');
    elseif loss==1
        [cs1,h1] = contour(XData,YData,Ploss1,[contour contour],'r');
        hold on;
        set(gca,'Tag','vrlcontaxes');
        [cs2,h2] = contour(XData,YData,Ploss2,[contour contour],'b');
        hold off;
        title('Contour plot of Power Losses');
    elseif ccl==1
        [cs1,h1] = contour(XData,YData,PVCCL1,[contour*10000 contour*10000],'r');
        hold on;
        set(gca,'Tag','vrlcontaxes');
        [cs2,h2] = contour(XData,YData,PVCCL2,[contour*10000 contour*10000],'b');
        hold off;
        title('Contour plot of LCC [MR]');
```

```
elseif ati==1
        [cs1,h1] = contour(XData,YData,ATI1,[contour contour],'r');
        hold on;
        set(gca,'Tag','vrlcontaxes');
        [cs2,h2] = contour(XData,YData,ATI2,[contour contour],'b');
        hold off;
        title('Conntour plot of ATI');
end;
set(gca,'Tag','vrlcontaxes');
clabel(cs1,h1)%,'labelspacing',100)
clabel(cs2,h2)%,'labelspacing',100)
ylabel('Distance [km]');
xlabel('Point Load [MW]');
legend(namec1,namec2,0);

if get(findobj(findobj('Tag','lfdcGUI'),'Tag','hrfcb'),'Value')
        fsize = 36;
        fig = findobj('Tag','2dvrlcontfig');
        if isempty(fig)
                figure
                if vr==1
                        set(gcf,'IntegerHandle','off',...
                        'Name','Voltage Regulation Contour Plot',...
                        'NumberTitle','off',...
                        'Tag','2dvrlcontfig');
                elseif loss==1
                        set(gcf,'IntegerHandle','off',...
                        'Name','Power Losses Contour Plot',...
                        'NumberTitle','off',...
                        'Tag','2dvrlcontfig');
                elseif ccl==1
                        set(gcf,'IntegerHandle','off',...
                        'Name','Cumulative Cost Contour Plot',...
                        'NumberTitle','off',...
                        'Tag','2dvrlcontfig');
                elseif ati==1
                        set(gcf,'IntegerHandle','off',...
                        'Name','ATI Contour Plot',...
                        'NumberTitle','off',...
                        'Tag','2dvrlcontfig');
                end;
        else
                figure(fig)
                if vr==1
                        set(gcf,'IntegerHandle','off',...
                        'Name','Voltage Regulation Contour Plot',...
                        'NumberTitle','off',...
                        'Tag','2dvrlcontfig');
                elseif loss==1
                        set(gcf,'IntegerHandle','off',...
                        'Name','Power Losses Contour Plot',...
                        'NumberTitle','off',...
                        'Tag','2dvrlcontfig');
                elseif ccl==1
                        set(gcf,'IntegerHandle','off',...
                        'Name','Cumulative Cost Contour Plot',...
                        'NumberTitle','off',...
                        'Tag','2dvrlcontfig');
                elseif ati==1
                        set(gcf,'IntegerHandle','off',...
                        'Name','ATI Contour Plot',...
                        'NumberTitle','off',...
                        'Tag','2dvrlcontfig');
                end;
        end;
        set(gcf,'Position',[1 29 1024 672]);
        set(gca,'Position',[0.15 0.15 0.82 0.82]);
        if vr==1
                [cs1,h] = contour(XData,YData,Vreg1,[contour contour],'r');
                set(h,'LineWidth',1.5);
                clabel(cs1,h);
                hold on;
                [cs2,h] = contour(XData,YData,Vreg2,[contour contour],'b');
                set(h,'LineWidth',1.5);
                clabel(cs2,h);
                hold off;
                ht=title('Contour plot of Voltage Regulation');
        elseif loss==1
                [cs1,h] = contour(XData,YData,Ploss1,[contour contour],'r');
                set(h,'LineWidth',1.5);
                clabel(cs1,h);
                hold on;
                [cs2,h] = contour(XData,YData,Ploss2,[contour contour],'b');
                set(h,'LineWidth',1.5);
                clabel(cs2,h);
                hold off;
                ht=title('Contour plot of Power Losses');
        elseif ccl==1
                [cs1,h] = contour(XData,YData,PVCCL1,[contour*10000 contour*10000],'r');
                set(h,'LineWidth',1.5);
                clabel(cs1,h);
                hold on;
                [cs2,h] = contour(XData,YData,PVCCL2,[contour*10000 contour*10000],'b');
                set(h,'LineWidth',1.5);
                clabel(cs2,h);
                hold off;
                ht=title('Contour plot of Cumulative Cost');
        elseif ati==1
                [cs1,h] = contour(XData,YData,ATI1,[contour contour],'r');
                set(h,'LineWidth',1.5);
                clabel(cs1,h);
```

```
                  hold on;
                  [cs2,h] = contour(XData,YData,ATI2,[contour contour],'b');
                  set(h,'LineWidth',1.5);
                  clabel(cs2,h);
                  hold off;
                  ht=title('Contour plot of ATI');
            end;
            hy=ylabel('Distance [km]');
            hx=xlabel('Point Load [MW]');
            set(hx,'FontSize',fsize*0.8,...
            'VerticalAlignment','top');
            set(hy,'FontSize',fsize*0.8,...
            'VerticalAlignment','bottom');
            set(ht,'FontSize',fsize);
            set(gcf,'Color',[1 1 1]);
            set(gca,'FontSize',fsize*0.7);
            axis tight
            legend(namec1,namec2,0);
      end;

elseif ~isempty(plotc1cont) & isempty(plotc2cont) & ~isempty(plotc3cont)
      axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrlcontaxes'))
      cla;
      if vr==1
            [cs1,h1] = contour(XData,YData,Vreg1,[contour contour],'r');
            hold on;
            set(gca,'Tag','vrlcontaxes');
            [cs3,h3] = contour(XData,YData,Vreg3,[contour contour],'g');
            hold off;
            title('Contour plot of Voltage Regulation');
      elseif loss==1
            [cs1,h1] = contour(XData,YData,Ploss1,[contour contour],'r');
            hold on;
            set(gca,'Tag','vrlcontaxes');
            [cs3,h3] = contour(XData,YData,Ploss3,[contour contour],'g');
            hold off;
            title('Contour plot of Power Losses');
      elseif ccl==1
            [cs1,h1] = contour(XData,YData,PVCCL1,[contour*10000 contour*10000],'r');
            hold on;
            set(gca,'Tag','vrlcontaxes');
            [cs3,h3] = contour(XData,YData,PVCCL3,[contour*10000 contour*10000],'g');
            hold off;
            title('Contour plot of LCC [MR]');
      elseif ati==1
            [cs1,h1] = contour(XData,YData,ATI1,[contour contour],'r');
            hold on;
            set(gca,'Tag','vrlcontaxes');
            [cs3,h3] = contour(XData,YData,ATI3,[contour contour],'g');
            hold off;
            title('Contour plot of ATI');
      end;
      set(gca,'Tag','vrlcontaxes');
      clabel(cs1,h1)%,'labelspacing',100)
      clabel(cs3,h3)%,'labelspacing',100)
      ylabel('Distance [km]');
      xlabel('Point Load [MW]');
      legend(namec1,namec3,0);

      if get(findobj(findobj('Tag','lfdcGUI'),'Tag','hrfcb'),'Value')
            fsize = 36;
            fig = findobj('Tag','2dvrlcontfig');
            if isempty(fig)
                  figure
                  if vr==1
                        set(gcf,'IntegerHandle','off',...
                        'Name','Voltage Regulation Contour Plot',...
                        'NumberTitle','off',...
                        'Tag','2dvrlcontfig');
                  elseif loss==1
                        set(gcf,'IntegerHandle','off',...
                        'Name','Power Losses Contour Plot',...
                        'NumberTitle','off',...
                        'Tag','2dvrlcontfig');
                  elseif ccl==1
                        set(gcf,'IntegerHandle','off',...
                        'Name','Cumulative Cost Contour Plot',...
                        'NumberTitle','off',...
                        'Tag','2dvrlcontfig');
                  elseif ati==1
                        set(gcf,'IntegerHandle','off',...
                        'Name','ATI Contour Plot',...
                        'NumberTitle','off',...
                        'Tag','2dvrlcontfig');
                  end;
            else
                  figure(fig)
                  if vr==1
                        set(gcf,'IntegerHandle','off',...
                        'Name','Voltage Regulation Contour Plot',...
                        'NumberTitle','off',...
                        'Tag','2dvrlcontfig');
                  elseif loss==1
                        set(gcf,'IntegerHandle','off',...
                        'Name','Power Losses Contour Plot',...
                        'NumberTitle','off',...
                        'Tag','2dvrlcontfig');
                  elseif ccl==1
                        set(gcf,'IntegerHandle','off',...
                        'Name','Cumulative Cost Contour Plot',...
                        'NumberTitle','off',...
```

```
                        'Tag','2dvrlcontfig');
                elseif ati==1
                        set(gcf,'IntegerHandle','off',...
                        'Name','ATI Contour Plot',...
                        'NumberTitle','off',...
                        'Tag','2dvrlcontfig');
                end;
            end;
            set(gcf,'Position',[1 29 1024 672]);
            set(gca,'Position',[0.15 0.15 0.82 0.82]);
            if vr==1
                [cs1,h] = contour(XData,YData,Vreg1,[contour contour],'r');
                set(h,'LineWidth',1.5);
                clabel(cs1,h);
                hold on;
                [cs3,h] = contour(XData,YData,Vreg3,[contour contour],'g');
                set(h,'LineWidth',1.5);
                clabel(cs3,h);
                hold off;
                ht=title('Contour plot of Voltage Regulation');
            elseif loss==1
                [cs1,h] = contour(XData,YData,Ploss1,[contour contour],'r');
                set(h,'LineWidth',1.5);
                clabel(cs1,h);
                hold on;
                [cs3,h] = contour(XData,YData,Ploss3,[contour contour],'g');
                set(h,'LineWidth',1.5);
                clabel(cs3,h);
                hold off;
                ht=title('Contour plot of Power Losses');
            elseif ccl==1
                [cs1,h] = contour(XData,YData,PVCCL1,[contour*10000 contour*10000],'r');
                set(h,'LineWidth',1.5);
                clabel(cs1,h);
                hold on;
                [cs3,h] = contour(XData,YData,PVCCL3,[contour*10000 contour*10000],'g');
                set(h,'LineWidth',1.5);
                clabel(cs3,h);
                hold off;
                ht=title('Contour plot of Cumulative Cost');
            elseif ati==1
                [cs1,h] = contour(XData,YData,ATI1,[contour contour],'r');
                set(h,'LineWidth',1.5);
                clabel(cs1,h);
                hold on;
                [cs3,h] = contour(XData,YData,ATI3,[contour contour],'g');
                set(h,'LineWidth',1.5);
                clabel(cs3,h);
                hold off;
                ht=title('Contour plot of ATI');
            end;
            hy=ylabel('Distance [km]');
            hx=xlabel('Point Load [MW]');
            set(hx,'FontSize',fsize*0.8,...
            'VerticalAlignment','top');
            set(hy,'FontSize',fsize*0.8,...
            'VerticalAlignment','bottom');
            set(ht,'FontSize',fsize);
            set(gcf,'Color',[1 1 1]);
            set(gca,'FontSize',fsize*0.7);
            axis tight
            legend(namec1,namec3,0);
        end;

    elseif isempty(plotc1cont) & ~isempty(plotc2cont) & ~isempty(plotc3cont)
        axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrlcontaxes'))
        cla;
        if vr==1
            [cs2,h2] = contour(XData,YData,Vreg2,[contour contour],'b');
            hold on;
            set(gca,'Tag','vrlcontaxes');
            [cs3,h3] = contour(XData,YData,Vreg3,[contour contour],'g');
            hold off;
            title('Contour plot of Voltage Regulation');
        elseif loss==1
            [cs2,h2] = contour(XData,YData,Ploss2,[contour contour],'b');
            hold on;
            set(gca,'Tag','vrlcontaxes');
            [cs3,h3] = contour(XData,YData,Ploss3,[contour contour],'g');
            hold off;
            title('Contour plot of Power Losses');
        elseif ccl==1
            [cs2,h2] = contour(XData,YData,PVCCL2,[contour*10000 contour*10000],'b');
            hold on;
            set(gca,'Tag','vrlcontaxes');
            [cs3,h3] = contour(XData,YData,PVCCL3,[contour*10000 contour*10000],'g');
            hold off;
            title('Contour plot of LCC [MR]');
        elseif ati==1
            [cs2,h2] = contour(XData,YData,ATI2,[contour contour],'b');
            hold on;
            set(gca,'Tag','vrlcontaxes');
            [cs3,h3] = contour(XData,YData,ATI3,[contour contour],'g');
            hold off;
            title('Contour plot of ATI');
        end;
        set(gca,'Tag','vrlcontaxes');
        clabel(cs2,h2)%,'labelspacing',100)
        clabel(cs3,h3)%,'labelspacing',100)
        ylabel('Distance [km]');
        xlabel('Point Load [MW]');
```

```
			legend(namec2,namec3,0);

			if get(findobj(findobj('Tag','lfdcGUI'),'Tag','hrfcb'),'Value')
				fsize = 36;
				fig = findobj('Tag','2dvrlcontfig');
				if isempty(fig)
					figure
					if vr==1
						set(gcf,'IntegerHandle','off',...
						'Name','Voltage Regulation Contour Plot',...
						'NumberTitle','off',...
						'Tag','2dvrlcontfig');
					elseif loss==1
						set(gcf,'IntegerHandle','off',...
						'Name','Power Losses Contour Plot',...
						'NumberTitle','off',...
						'Tag','2dvrlcontfig');
					elseif ccl==1
						set(gcf,'IntegerHandle','off',...
						'Name','Cumulative Cost Contour Plot',...
						'NumberTitle','off',...
						'Tag','2dvrlcontfig');
					elseif ati==1
						set(gcf,'IntegerHandle','off',...
						'Name','ATI Contour Plot',...
						'NumberTitle','off',...
						'Tag','2dvrlcontfig');
					end;
				else
					figure(fig)
					if vr==1
						set(gcf,'IntegerHandle','off',...
						'Name','Voltage Regulation Contour Plot',...
						'NumberTitle','off',...
						'Tag','2dvrlcontfig');
					elseif loss==1
						set(gcf,'IntegerHandle','off',...
						'Name','Power Losses Contour Plot',...
						'NumberTitle','off',...
						'Tag','2dvrlcontfig');
					elseif ccl==1
						set(gcf,'IntegerHandle','off',...
						'Name','Cumulative Cost Contour Plot',...
						'NumberTitle','off',...
						'Tag','2dvrlcontfig');
					elseif ati==1
						set(gcf,'IntegerHandle','off',...
						'Name','ATI Contour Plot',...
						'NumberTitle','off',...
						'Tag','2dvrlcontfig');
					end;
				end;
				set(gcf,'Position',[1 29 1024 672]);
				set(gca,'Position',[0.15 0.15 0.82 0.82]);
				if vr==1
					[cs2,h] = contour(XData,YData,Vreg2,[contour contour],'b');
					set(h,'LineWidth',1.5);
					clabel(cs2,h);
					hold on;
					[cs3,h] = contour(XData,YData,Vreg3,[contour contour],'g');
					set(h,'LineWidth',1.5);
					clabel(cs3,h);
					hold off;
					ht=title('Contour plot of Voltage Regulation');
				elseif loss==1
					[cs2,h] = contour(XData,YData,Ploss2,[contour contour],'b');
					set(h,'LineWidth',1.5);
					clabel(cs2,h);
					hold on;
					[cs3,h] = contour(XData,YData,Ploss3,[contour contour],'g');
					set(h,'LineWidth',1.5);
					clabel(cs3,h);
					hold off;
					ht=title('Contour plot of Power Losses');
				elseif ccl==1
					[cs2,h] = contour(XData,YData,PVCCL2,[contour*10000 contour*10000],'b');
					set(h,'LineWidth',1.5);
					clabel(cs2,h);
					hold on;
					[cs3,h] = contour(XData,YData,PVCCL3,[contour*10000 contour*10000],'g');
					set(h,'LineWidth',1.5);
					clabel(cs3,h);
					hold off;
					ht=title('Contour plot of Cumulative Cost');
				elseif ati==1
					[cs2,h] = contour(XData,YData,ATI2,[contour contour],'b');
					set(h,'LineWidth',1.5);
					clabel(cs2,h);
					hold on;
					[cs3,h] = contour(XData,YData,ATI3,[contour contour],'g');
					set(h,'LineWidth',1.5);
					clabel(cs3,h);
					hold off;
					ht=title('Contour plot of ATI');
				end;
				hy=ylabel('Distance [km]');
				hx=xlabel('Point Load [MW]');
				set(hx,'FontSize',fsize*0.8,...
				'VerticalAlignment','top');
				set(hy,'FontSize',fsize*0.8,...
				'VerticalAlignment','bottom');
```

```
                set(ht,'FontSize',fsize);
                set(gcf,'Color',[1 1 1]);
                set(gca,'FontSize',fsize*0.7);
                axis tight
                legend(namec2,namec3,0);
            end;

        elseif ~isempty(plotc1cont) & isempty(plotc2cont) & isempty(plotc3cont)
            axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrlcontaxes'))
            cla;
            if vr==1
                [cs1,h1] = contour(XData,YData,Vreg1,[contour contour],'r');
                title('Contour plot of Voltage Regulation');
            elseif loss==1
                [cs1,h1] = contour(XData,YData,Ploss1,[contour contour],'r');
                title('Contour plot of Power Losses');
            elseif ccl==1
                [cs1,h1] = contour(XData,YData,PVCCL1,[contour*10000 contour*10000],'r');
                title('Contour plot of LCC [MR]');
            elseif ati==1
                [cs1,h1] = contour(XData,YData,ATI1,[contour contour],'r');
                title('Contour plot of ATI');
            end;
            set(gca,'Tag','vrlcontaxes');
            clabel(cs1,h1)%,'labelspacing',100)
            ylabel('Distance [km]');
            xlabel('Point Load [MW]');
            legend(namec1,0);

            if get(findobj(findobj('Tag','lfdcGUI'),'Tag','hrfcb'),'Value')
                fsize = 36;
                fig = findobj('Tag','2dvrlcontfig');
                if isempty(fig)
                    figure
                    if vr==1
                        set(gcf,'IntegerHandle','off',...
                        'Name','Voltage Regulation Contour Plot',...
                        'NumberTitle','off',...
                        'Tag','2dvrlcontfig');
                    elseif loss==1
                        set(gcf,'IntegerHandle','off',...
                        'Name','Power Losses Contour Plot',...
                        'NumberTitle','off',...
                        'Tag','2dvrlcontfig');
                    elseif ccl==1
                        set(gcf,'IntegerHandle','off',...
                        'Name','Cumulative Cost Contour Plot',...
                        'NumberTitle','off',...
                        'Tag','2dvrlcontfig');
                    elseif ati==1
                        set(gcf,'IntegerHandle','off',...
                        'Name','ATI Contour Plot',...
                        'NumberTitle','off',...
                        'Tag','2dvrlcontfig');
                     end;
                else
                    figure(fig)
                    if vr==1
                        set(gcf,'IntegerHandle','off',...
                        'Name','Voltage Regulation Contour Plot',...
                        'NumberTitle','off',...
                        'Tag','2dvrlcontfig');
                    elseif loss==1
                        set(gcf,'IntegerHandle','off',...
                        'Name','Power Losses Contour Plot',...
                        'NumberTitle','off',...
                        'Tag','2dvrlcontfig');
                    elseif ccl==1
                        set(gcf,'IntegerHandle','off',...
                        'Name','Cumulative Cost Contour Plot',...
                        'NumberTitle','off',...
                        'Tag','2dvrlcontfig');
                    elseif ati==1
                        set(gcf,'IntegerHandle','off',...
                        'Name','ATI Contour Plot',...
                        'NumberTitle','off',...
                        'Tag','2dvrlcontfig'
                    end;
                end;
                set(gcf,'Position',[1 29 1024 672]);
                set(gca,'Position',[0.15 0.15 0.82 0.82]);
                if vr==1
                    [cs1,h] = contour(XData,YData,Vreg1,[contour contour],'r');
                    set(h,'LineWidth',1.5);
                    clabel(cs1,h);
                    ht=title('Contour plot of Voltage Regulation');
                elseif loss==1
                    [cs1,h] = contour(XData,YData,Ploss1,[contour contour],'r');
                    set(h,'LineWidth',1.5);
                    clabel(cs1,h);
                    ht=title('Contour plot of Power Losses');
                elseif ccl==1
                    [cs1,h] = contour(XData,YData,PVCCL1,[contour*10000 contour*10000],'r');
                    set(h,'LineWidth',1.5);
                    clabel(cs1,h);
                    ht=title('Contour plot of Cumulative Cost');
                elseif ati==1
                    [cs1,h] = contour(XData,YData,ATI1,[contour contour],'r');
                    set(h,'LineWidth',1.5);
                    clabel(cs1,h);
                    ht=title('Contour plot of ATI');
```

```
				end;
				hy=ylabel('Distance [km]');
				hx=xlabel('Point Load [MW]');
				set(hx,'FontSize',fsize*0.8,...
				'VerticalAlignment','top');
				set(hy,'FontSize',fsize*0.8,...
				'VerticalAlignment','bottom');
				set(ht,'FontSize',fsize);
				set(gcf,'Color',[1 1 1]);
				set(gca,'FontSize',fsize*0.7);
				axis tight
				legend(namec1,0);
			end;

		elseif isempty(plotc1cont) & ~isempty(plotc2cont) & isempty(plotc3cont)
			axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrlcontaxes'))
			cla;
			if vr==1
				[cs2,h2] = contour(XData,YData,Vreg2,[contour contour],'b');
				title('Contour plot of Voltage Regulation');
			elseif loss==1
				[cs2,h2] = contour(XData,YData,Ploss2,[contour contour],'b');
				title('Contour plot of Power Losses');
			elseif ccl==1
				[cs2,h2] = contour(XData,YData,PVCCL2,[contour*10000 contour*10000],'b');
				title('Contour plot of LCC [MR]');
			elseif ati==1
				[cs2,h2] = contour(XData,YData,ATI2,[contour contour],'b');
				title('Contour plot of ATI');
			end;
			set(gca,'Tag','vrlcontaxes');
			clabel(cs2,h2)%,'labelspacing',100)
			ylabel('Distance [km]');
			xlabel('Point Load [MW]');
			legend(namec2,0);

			if get(findobj(findobj('Tag','lfdcGUI'),'Tag','hrfcb'),'Value')
				fsize = 36;
				fig = findobj('Tag','2dvrlcontfig');
				if isempty(fig)
					figure
					if vr==1
						set(gcf,'IntegerHandle','off',...
						'Name','Voltage Regulation Contour Plot',...
						'NumberTitle','off',...
						'Tag','2dvrlcontfig');
					elseif loss==1
						set(gcf,'IntegerHandle','off',...
						'Name','Power Losses Contour Plot',...
						'NumberTitle','off',...
						'Tag','2dvrlcontfig');
					elseif ccl==1
						set(gcf,'IntegerHandle','off',...
						'Name','Cumulative Cost Contour Plot',...
						'NumberTitle','off',...
						'Tag','2dvrlcontfig');
					elseif ati==1
						set(gcf,'IntegerHandle','off',...
						'Name','ATI Contour Plot',...
						'NumberTitle','off',...
						'Tag','2dvrlcontfig');
					end;
				else
					figure(fig)
					if vr==1
						set(gcf,'IntegerHandle','off',...
						'Name','Voltage Regulation Contour Plot',...
						'NumberTitle','off',...
						'Tag','2dvrlcontfig');
					elseif loss==1
						set(gcf,'IntegerHandle','off',...
						'Name','Power Losses Contour Plot',...
						'NumberTitle','off',...
						'Tag','2dvrlcontfig');
					elseif ccl==1
						set(gcf,'IntegerHandle','off',...
						'Name','Cumulative Cost Contour Plot',...
						'NumberTitle','off',...
						'Tag','2dvrlcontfig');
					elseif ati==1
						set(gcf,'IntegerHandle','off',...
						'Name','ATI Contour Plot',...
						'NumberTitle','off',...
						'Tag','2dvrlcontfig');
					end;
				end;
				set(gcf,'Position',[1 29 1024 672]);
				set(gca,'Position',[0.15 0.15 0.82 0.82]);
				if vr==1
					[cs2,h] = contour(XData,YData,Vreg2,[contour contour],'b');
					set(h,'LineWidth',1.5);
					clabel(cs2,h);
					ht=title('Contour plot of Voltage Regulation');
				elseif loss==1
					[cs2,h] = contour(XData,YData,Ploss2,[contour contour],'b');
					set(h,'LineWidth',1.5);
					clabel(cs2,h);
					ht=title('Contour plot of Power Losses');
				elseif ccl==1
					[cs2,h] = contour(XData,YData,PVCCL2,[contour*10000 contour*10000],'b');
					set(h,'LineWidth',1.5);
```

```
                        clabel(cs2,h);
                        ht=title('Contour plot of Cumulative Cost');
                elseif ati==1
                        [cs2,h] = contour(XData,YData,ATI2,[contour contour],'b');
                        set(h,'LineWidth',1.5);
                        clabel(cs2,h);
                        ht=title('Contour plot of ATI');
                end;
                hy=ylabel('Distance [km]');
                hx=xlabel('Point Load [MW]');
                set(hx,'FontSize',fsize*0.8,...
                'VerticalAlignment','top');
                set(hy,'FontSize',fsize*0.8,...
                'VerticalAlignment','bottom');
                set(ht,'FontSize',fsize);
                set(gcf,'Color',[1 1 1]);
                set(gca,'FontSize',fsize*0.7);
                axis tight
                legend(namec2,0);
        end;

elseif isempty(plotc1cont) & isempty(plotc2cont) & ~isempty(plotc3cont)
        axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrlcontaxes'))
        cla;
        if vr==1
                [cs3,h3] = contour(XData,YData,Vreg3,[contour contour],'g');
                title('Contour plot of Voltage Regulation');
        elseif loss==1
                [cs3,h3] = contour(XData,YData,Ploss3,[contour contour],'g');
                title('Contour plot of Power Losses');
        elseif ccl==1
                [cs3,h3] = contour(XData,YData,PVCCL3,[contour*10000 contour*10000],'g');
                title('Contour plot of LCC [MR]');
        elseif ati==1
                [cs3,h3] = contour(XData,YData,ATI3,[contour contour],'g');
                title('Contour plot of ATI');
        end;
        set(gca,'Tag','vrlcontaxes');
        clabel(cs3,h3)%,'labelspacing',100)
        ylabel('Distance [km]');
        xlabel('Point Load [MW]');
        legend(namec3,0);

        if get(findobj(findobj('Tag','lfdcGUI'),'Tag','hrfcb'),'Value')
                fsize = 36;
                fig = findobj('Tag','2dvrlcontfig');
                if isempty(fig)
                        figure
                        if vr==1
                                set(gcf,'IntegerHandle','off',...
                                'Name','Voltage Regulation Contour Plot',...
                                'NumberTitle','off',...
                                'Tag','2dvrlcontfig');
                        elseif loss==1
                                set(gcf,'IntegerHandle','off',...
                                'Name','Power Losses Contour Plot',...
                                'NumberTitle','off',...
                                'Tag','2dvrlcontfig');
                        elseif ccl==1
                                set(gcf,'IntegerHandle','off',...
                                'Name','LCC Contour Plot',...
                                'NumberTitle','off',...
                                'Tag','2dvrlcontfig');
                        elseif ati==1
                                set(gcf,'IntegerHandle','off',...
                                'Name','ATI Contour Plot',...
                                'NumberTitle','off',...
                                'Tag','2dvrlcontfig');
                        end;
                else
                        figure(fig)
                        if vr==1
                                set(gcf,'IntegerHandle','off',...
                                'Name','Voltage Regulation Contour Plot',...
                                'NumberTitle','off',...
                                'Tag','2dvrlcontfig');
                        elseif loss==1
                                set(gcf,'IntegerHandle','off',...
                                'Name','Power Losses Contour Plot',...
                                'NumberTitle','off',...
                                'Tag','2dvrlcontfig');
                        elseif ccl==1
                                set(gcf,'IntegerHandle','off',...
                                'Name','LCC Contour Plot',...
                                'NumberTitle','off',...
                                'Tag','2dvrlcontfig');
                        elseif ati==1
                                set(gcf,'IntegerHandle','off',...
                                'Name','ATI Contour Plot',...
                                'NumberTitle','off',...
                                'Tag','2dvrlcontfig');
                        end;
                end;
                set(gcf,'Position',[1 29 1024 672]);
                set(gca,'Position',[0.15 0.15 0.82 0.82]);
                if vr==1
                        [cs3,h] = contour(XData,YData,Vreg3,[contour contour],'g');
                        set(h,'LineWidth',1.5);
                        clabel(cs3,h);
                        ht=title('Contour plot of Voltage Regulation');
                elseif loss==1
```

```
                [cs3,h] = contour(XData,YData,Ploss3,[contour contour],'g');
                set(h,'LineWidth',1.5);
                clabel(cs3,h);
                ht=title('Contour plot of Power Losses');
            elseif ccl==1
                [cs3,h] = contour(XData,YData,PVCCL3,[contour*10000 contour*10000],'g');
                set(h,'LineWidth',1.5);
                clabel(cs3,h);
                ht=title('Contour plot of LCC');
            elseif ati==1
                [cs3,h] = contour(XData,YData,ATI3,[contour contour],'g');
                set(h,'LineWidth',1.5);
                clabel(cs3,h);
                ht=title('Contour plot of ATI');
            end;
            hy=ylabel('Distance [km]');
            hx=xlabel('Point Load [MW]');
            set(hx,'FontSize',fsize*0.8,...
            'VerticalAlignment','top');
            set(hy,'FontSize',fsize*0.8,...
            'VerticalAlignment','bottom');
            set(ht,'FontSize',fsize);
            set(gcf,'Color',[1 1 1]);
            set(gca,'FontSize',fsize*0.7);
            axis tight
            legend(namec3,0);
        end;

elseif isempty(plotc1cont) & isempty(plotc2cont) & isempty(plotc3cont)
        axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrlcontaxes'))
        cla;
        if vr==1
            title('Contour plot of Voltage Regulation');
        elseif loss==1
            title('Contour plot of Power Losses');
        elseif ccl==1
            title('Contour plot of LCC');
        elseif ati==1
            title('Contour plot of ATI');
        end;
        set(gca,'Tag','vrlcontaxes');

        if get(findobj(findobj('Tag','lfdcGUI'),'Tag','hrfcb'),'Value')
            fsize = 36;
            fig = findobj('Tag','2dvrlcontfig');
            if isempty(fig)
                figure
                if vr==1
                    set(gcf,'IntegerHandle','off',...
                    'Name','Voltage Regulation Contour Plot',...
                    'NumberTitle','off',...
                    'Tag','2dvrlcontfig');
                elseif loss==1
                    set(gcf,'IntegerHandle','off',...
                    'Name','Power Losses Contour Plot',...
                    'NumberTitle','off',...
                    'Tag','2dvrlcontfig');
                elseif ccl==1
                    set(gcf,'IntegerHandle','off',...
                    'Name','LCC Contour Plot',...
                    'NumberTitle','off',...
                    'Tag','2dvrlcontfig');
                elseif ati==1
                    set(gcf,'IntegerHandle','off',...
                    'Name','ATI Contour Plot',...
                    'NumberTitle','off',...
                    'Tag','2dvrlcontfig'
                end;
            else
                figure(fig)
                if vr==1
                    set(gcf,'IntegerHandle','off',...
                    'Name','Voltage Regulation Contour Plot',...
                    'NumberTitle','off',...
                    'Tag','2dvrlcontfig');
                elseif loss==1
                    set(gcf,'IntegerHandle','off',...
                    'Name','Power Losses Contour Plot',...
                    'NumberTitle','off',...
                    'Tag','2dvrlcontfig');
                elseif ccl==1
                    set(gcf,'IntegerHandle','off',...
                    'Name','LCC Contour Plot',...
                    'NumberTitle','off',...
                    'Tag','2dvrlcontfig');
                elseif ati==1
                    set(gcf,'IntegerHandle','off',...
                    'Name','ATI Contour Plot',...
                    'NumberTitle','off',...
                    'Tag','2dvrlcontfig');
                end;
            end;
            set(gcf,'Position',[1 29 1024 672]);
            set(gca,'Position',[0.15 0.15 0.82 0.82]);
            if vr==1
                cla;
                ht=title('Contour plot of Voltage Regulation');
            elseif loss==1
                cla;
                ht=title('Contour plot of Power Losses');
            elseif ccl==1
```

```
                cla;
                ht=title('Contour plot of LCC');
            elseif ati==1
                cla;
                ht=title('Contour plot of ATI');
            end;
            hy=ylabel('Distance [km]');
            hx=xlabel('Point Load [MW]');
            set(hx,'FontSize',fsize*0.8,...
            'VerticalAlignment','top');
            set(hy,'FontSize',fsize*0.8,...
            'VerticalAlignment','bottom');
            set(ht,'FontSize',fsize);
            set(gcf,'Color',[1 1 1]);
            set(gca,'FontSize',fsize*0.7);
        end;
    end;
```

B.4.4 Plot3d.m

```
function plot3d(XData,YData,namec1,namec2,namec3,UserData,Data)

    vr      =   UserData(23);
    loss    =   UserData(24);
    ccl     =   UserData(25);
    ati     =   UserData(26);

    Vreg1   =   Data(:,:,1);
    Vreg2   =   Data(:,:,2);
    Vreg3   =   Data(:,:,3);
    Ploss1  =   Data(:,:,4);
    Ploss2  =   Data(:,:,5);
    Ploss3  =   Data(:,:,6);
    PVCCL1  =   Data(:,:,7);
    PVCCL2  =   Data(:,:,8);
    PVCCL3  =   Data(:,:,9);
    ATI1    =   Data(:,:,10);
    ATI2    =   Data(:,:,11);
    ATI3    =   Data(:,:,12);

    XData1  =   XData(:,:,1);
    XData2  =   XData(:,:,2);
    XData3  =   XData(:,:,3);

    axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vr1laxes'))
    cla;
    if vr==1
        surf(XData1,YData,Vreg1);
        zlabel('Voltage Regulation [ %] ');
    elseif loss==1
        surf(XData1,YData,Ploss1);
        zlabel('Power Losses [ kW] ');
    elseif ccl==1
        surf(XData1,YData,PVCCL1);
        zlabel('Cumulative Cost [ MR] ');
    elseif ati==1
        surf(XData1,YData,ATI1);
        zlabel('ATI');
    end;
    set(gca,'Tag','vr1laxes');
    view([ 60 14] )
    shading interp
    ylabel('Distance [ km] ');
    xlabel('Point Load [ MW] ');
    title (namec1);

    axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrl2axes'))
    cla;
    if vr==1
        surf(XData2,YData,Vreg2);
        zlabel('Voltage Regulation [ %] ');
    elseif loss==1
        surf(XData2,YData,Ploss2);
        zlabel('Power Losses [ kW] ');
    elseif ccl==1
        surf(XData2,YData,PVCCL2);
        zlabel('Cumulative Cost [ MR] ');
    elseif ati==1
        surf(XData2,YData,ATI2);
        zlabel('ATI');
    end;
    set(gca,'Tag','vrl2axes');
    view([ 60 14] )
    shading interp
    ylabel('Distance [ km] ');
    xlabel('Point Load [ MW] ');
    title (namec2);

    axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrl3axes'))
    cla;
    if vr==1
        surf(XData3,YData,Vreg3);
        zlabel('Voltage Regulation [ %] ');
    elseif loss==1
        surf(XData3,YData,Ploss3);
        zlabel('Power Losses [ kW] ');
    elseif ccl==1
        surf(XData3,YData,PVCCL3);
        zlabel('Cumulative Cost [ MR] ');
    elseif ati==1
        surf(XData3,YData,ATI3);
        zlabel('ATI');
    end;
    set(gca,'Tag','vrl3axes');
    view([ 60 14] )
    shading interp
    ylabel('Distance [ km] ');
    xlabel('Point Load [ MW] ');
    title (namec3);

    if get(findobj(findobj('Tag','lfdcGUI'),'Tag','hrfcb'),'Value')
        fsize = 36;
        fig = findobj('Tag','3dvrlc1fig');
        if isempty(fig)
            figure
            if vr==1
                set(gcf,'IntegerHandle','off',...
```

```
            'Name','Voltage Regulation',...
            'NumberTitle','off',...
            'Tag','3dvrlc1fig');
        elseif loss==1
            set(gcf,'IntegerHandle','off',...
            'Name','Power Losses ',...
            'NumberTitle','off',...
            'Tag','3dvrlc1fig');
        elseif ccl==1
            set(gcf,'IntegerHandle','off',...
            'Name','LCC',...
            'NumberTitle','off',...
            'Tag','3dvrlc1fig');
        elseif ati==1
            set(gcf,'IntegerHandle','off',...
            'Name','ATI',...
            'NumberTitle','off',...
            'Tag','3dvrlc1fig');
        end;
    else
        figure(fig)
        if vr==1
            set(gcf,'IntegerHandle','off',...
            'Name','Voltage Regulation',...
            'NumberTitle','off',...
            'Tag','3dvrlc1fig');
        elseif loss==1
            set(gcf,'IntegerHandle','off',...
            'Name','Power Losses ',...
            'NumberTitle','off',...
            'Tag','3dvrlc1fig');
        elseif ccl==1
            set(gcf,'IntegerHandle','off',...
            'Name','LCC',...
            'NumberTitle','off',...
            'Tag','3dvrlc1fig');
        elseif ati==1
            set(gcf,'IntegerHandle','off',...
            'Name','ATI',...
            'NumberTitle','off',...
            'Tag','3dvrlc1fig');
        end;
end;

set(gcf,'Position',[1 29 1024 672]);
set(gca,'Position',[0.17 0.12 0.7 0.75]);
if vr==1
    surfH=surf(XData1,YData,Vreg1);
    hz=zlabel('Voltage Regulation [%]');
elseif loss==1
    surfH=surf(XData1,YData,Ploss1);
    hz=zlabel('Power Losses [kW]');
elseif ccl==1
    surfH=surf(XData1,YData,PVCCL1);
    hz=zlabel('Cumulative Cost [MR]');
elseif ati==1
    surfH=surf(XData1,YData,ATI1);
    hz=zlabel('ATI');
end;

ht=title (namecl);
h=xlabel('Point Load [MW]');
set(h,'FontSize',fsize*0.8,...
'VerticalAlignment','middle');
h=ylabel('Distance [km]');
set(h,'FontSize',fsize*0.8,...
'VerticalAlignment','cap');
set(ht,'FontSize',fsize);
posz = get(hz,'Position');
set(hz,'FontSize',fsize*0.8,...
'VerticalAlignment','bottom');
set(gcf,'Color',[1 1 1]);
set(gca,'FontSize',fsize*0.7);

shading interp
currAxisLim = axis;
axis([currAxisLim(1) currAxisLim(2) currAxisLim(3) currAxisLim(4) currAxisLim(5) currAxisLim(6)])
view(60,14)

fig = findobj('Tag','3dvrlc2fig');
if isempty(fig)
    figure
    if vr==1
        set(gcf,'IntegerHandle','off',...
        'Name','Voltage Regulation',...
        'NumberTitle','off',...
        'Tag','3dvrlc2fig');
    elseif loss==1
        set(gcf,'IntegerHandle','off',...
        'Name','Power Losses ',...
        'NumberTitle','off',...
        'Tag','3dvrlc2fig');
    elseif ccl==1
        set(gcf,'IntegerHandle','off',...
        'Name','LCC',...
        'NumberTitle','off',...
        'Tag','3dvrlc2fig');
    elseif ati==1
        set(gcf,'IntegerHandle','off',...
        'Name','ATI',...
        'NumberTitle','off',...
        'Tag','3dvrlc2fig');
```

```
        end;
    else
        figure(fig)
        if vr==1
            set(gcf,'IntegerHandle','off',...
            'Name','Voltage Regulation',...
            'NumberTitle','off',...
            'Tag','3dvrlc2fig');
        elseif loss==1
            set(gcf,'IntegerHandle','off',...
            'Name','Power Losses ',...
            'NumberTitle','off',...
            'Tag','3dvrlc2fig');
        elseif ccl==1
            set(gcf,'IntegerHandle','off',...
            'Name','LCC',...
            'NumberTitle','off',...
            'Tag','3dvrlc2fig');
        elseif ati==1
            set(gcf,'IntegerHandle','off',...
            'Name','ATI',...
            'NumberTitle','off',...
            'Tag','3dvrlc2fig');
        end;
    end;
    set(gcf,'Position',[ 1 29 1024 672] );
    set(gca,'Position',[ 0.17 0.12 0.7 0.75] );
    if vr==1
        surfH=surf(XData2,YData,Vreg2);
        hz=zlabel('Voltage Regulation [ %] ');
    elseif loss==1
        surfH=surf(XData2,YData,Ploss2);
        hz=zlabel('Power Losses [ kW] ');
    elseif ccl==1
        surfH=surf(XData2,YData,PVCCL2);
        hz=zlabel('Cumulative Cost [ MR] ');
    elseif ati==1
        surfH=surf(XData2,YData,ATI2);
        hz=zlabel('ATI');
    end;

    ht=title (namec2);
    h=xlabel('Point Load [ MW] ');
    set(h,'FontSize',fsize*0.8,...
    'VerticalAlignment','middle');
    h=ylabel('Distance [ km] ');
    set(h,'FontSize',fsize*0.8,...
    'VerticalAlignment','cap');
    set(ht,'FontSize',fsize);
    posz = get(hz,'Position');
    set(hz,'FontSize',fsize*0.8,...
    'VerticalAlignment','bottom');
    set(gcf,'Color',[ 1 1 1] );
    set(gca,'FontSize',fsize*0.7);

    shading interp
    currAxisLim = axis;
    axis([ currAxisLim(1) currAxisLim(2) currAxisLim(3) currAxisLim(4) currAxisLim(5) currAxisLim(6)] )
    view(60,14)

    fig = findobj('Tag','3dvrlc3fig');
    if isempty(fig)
        figure
        if vr==1
            set(gcf,'IntegerHandle','off',...
            'Name','Voltage Regulation',...
            'NumberTitle','off',...
            'Tag','3dvrlc3fig');
        elseif loss==1
            set(gcf,'IntegerHandle','off',...
            'Name','Power Losses ',...
            'NumberTitle','off',...
            'Tag','3dvrlc3fig');
        elseif ccl==1
            set(gcf,'IntegerHandle','off',...
            'Name','LCC',...
            'NumberTitle','off',...
            'Tag','3dvrlc3fig');
        elseif ati==1
            set(gcf,'IntegerHandle','off',...
            'Name','ATI',...
            'NumberTitle','off',...
            'Tag','3dvrlc3fig');
        end;
    else
        figure(fig)
        if vr==1
            set(gcf,'IntegerHandle','off',...
            'Name','Voltage Regulation',...
            'NumberTitle','off',...
            'Tag','3dvrlc3fig');
        elseif loss==1
            set(gcf,'IntegerHandle','off',...
            'Name','Power Losses ',...
            'NumberTitle','off',...
            'Tag','3dvrlc3fig');
        elseif ccl==1
            set(gcf,'IntegerHandle','off',...
            'Name','LCC',...
            'NumberTitle','off',...
            'Tag','3dvrlc3fig');
        elseif ati==1
```

```
                    set(gcf,'IntegerHandle','off',...
                    'Name','ATI',...
                    'NumberTitle','off',...
                    'Tag','3dvrlc3fig');
                end;
            end;
            set(gcf,'Position',[ 1 29 1024 672] );
            set(gca,'Position',[ 0.17 0.12 0.7 0.75] );
            if vr==1
                surfH=surf(XData3,YData,Vreg3);
                hz=zlabel('Voltage Regulation [ %] ');
            elseif loss==1
                surfH=surf(XData3,YData,Ploss3);
                hz=zlabel('Power Losses [ kW] ');
            elseif ccl==1
                surfH=surf(XData3,YData,PVCCL3);
                hz=zlabel('Cumulative Cost [ MR] ');
            elseif ati==1
                surfH=surf(XData3,YData,ATI3);
                hz=zlabel('ATI');
            end;

            ht=title (namec3);
            h=xlabel('Point Load [ MW] ');
            set(h,'FontSize',fsize*0.8,...
            'VerticalAlignment','middle');
            h=ylabel('Distance [ km] ');
            set(h,'FontSize',fsize*0.8,...
            'VerticalAlignment','cap');
            set(ht,'FontSize',fsize);
            posz = get(hz,'Position');
            set(hz,'FontSize',fsize*0.8,...
            'VerticalAlignment','bottom');
            set(gcf,'Color',[ 1 1 1] );
            set(gca,'FontSize',fsize*0.7);

            shading interp
            currAxisLim = axis;
            axis([ currAxisLim(1) currAxisLim(2) currAxisLim(3) currAxisLim(4) currAxisLim(5) currAxisLim(6)] )
            view(60,14)
        end;
```

B.5 FUNCTION CODE TO CALCULATE THE THERMAL LOAD REACH OF THREE-PHASE LINES

B.5.1 Constants.m

```
%Matlab routine to calculate the thermal load reach of different three-phase line technologies

%Wolf Conductor
S=(86.42e6);                              %Determenistic or Probabilistic Thermal Load
pf=0.95;                                  %power factor
P=(S*pf);                                 %real power
Q=(sin(acos(pf))*S);                      %reacive power


r=0.2091;                                 %resistance/km
x=0.4157;                                 %reactance/km
b=2.72e-6;                                %supceptance/km

Vs=132e3;                                 %sending-end voltage
vd=0.05;                                  %per unit voltage drop
Vr=Vs-Vs*vd;                              %receiving-end voltage

%Calculate constants
c1=Vr^4*b^2*(r^2+x^2);
c2=-4*Q*Vr^2*b*(r^2-b^2);
c3=4*(S^2*r^2+S^2*x^2-Vr^4*b*x);
c4=8*Vr^2*(r*P+x*Q);
c5=4*Vr^2*(Vr^2-Vs^2);

%Constant matrix
c=[c1 c2 c3 c4 c5];

%Solve matrix for all possible roots
roots(c)
```

B.6 FUNCTION CODE TO CALCULATE THE TRANSFER CAPABILITIES OF THREE-PHASE LINES

B.6.1 Transfer.m

```
%Wolf Conductor
pf=0.95;                                    %power factor

r=0.2091;                                   %resistance/km
x=0.4157;                                   %reactance/km
b=2.72e-6;                                  %supceptance/km
z2=(r^2 + x^2);

Vs=132e3;                                   %sending-end voltage
vd=0.05;                                    %per unit voltage drop
Vr=(Vs-Vs*vd);                              %receiving-end voltage

%Calculate constants
c1=z2;
c2=(2*r*pf + (sin(acos(pf)))*(2*x-b*z2))*Vr^2;
c3=(1-b*x+(b^2*z2)/4)*Vr^4 - Vs^2*Vr^2;

transfercapability=[c1 c2 c3];

tc=(roots(transfercapability))
```